# Evaluating Quality of Online Behavior Data

*By Marcus Berg*

*Supervised by Lars Lyberg*

Master's Thesis
Department of Statistics
Stockholm University
June 2013

**Abstract**

This thesis has two purposes; emphasizing the importance of data quality of Big Data, and identifying and evaluating potential error sources in JavaScript tracking (a client side on-site online behavior clickstream data collection method commonly used in web analytics). The importance of data quality of Big Data is emphasized through the evaluation of JavaScript tracking. The Total Survey Error framework is applied to JavaScript tracking and 17 nonsampling error sources are identified and evaluated. The bias imposed by these error sources varies from large to small, but the major takeaway is the large number of error sources actually identified. More work is needed. Big Data has much to gain from quality work. Similarly, there is much that can be done with statistics in web analytics.

**Acknowledgements**

I would like to thank David Rasmusson, Gordon Savage and Mattias Ward for introducing me to web tracking and web analytics.

And another thank you to Åsa Jonsson for cheering me on in this endeavor.

## Table of Contents

# Introduction

Big Data has been heralded as the next great data revolution and is becoming one of the cornerstones of modern business development (Champkin 2012a, Manyika et al. 2011). Large volumes of data are collected, stored and connected in a continuously growing network of systems and data repositories (Keller et al 2012). Data-driven organizations use this data to manage their business development, preferably in real time (Rencher 2013). Yet, it is important not to get blinded by the immense possibilities that Big Data is offering and to remember that the results of any analysis is at its core dependent on the quality of the data that it is based on in the first place, something that has been advocated lately by leading companies in the market (Mastrangelo 2012).

Simultaneous with the advent of Big Data, the Internet has grown as a channel for commerce, and companies are turning to the web analytics industry to gain insight into how their customers are behaving and how to interact with them online. Using a JavaScript tracking solution, companies and organizations can collect data on their visitors' behavior and get answers to questions such as: 'Do our customers find the information that they are looking for on our site?', 'What do our customers do on our site before they purchase any of our products?' and 'Do our customers have any problems trying to buy the products on our site?' (Kaushik 2009). By understanding how their visitors move around the website, how they interact with the pages, and what equipment they use, companies and organizations can redesign their websites to better fit their customers' and their own needs.

As with Big Data, analysis of online behavior data is at its core dependent on the quality of the data that the analysis is based on. It is not enough to assure that the data is collected and stored without incident. Data accuracy needs to be evaluated as well. Biemer (2010) concludes that survey users and other stakeholders often take accuracy for granted, and there are reasons to believe that the situation is the same regarding web tracking data. Thus there is a need for evaluating the quality of data that has been collected through JavaScript tracking.

Total Survey Error (TSE) is a statistical framework for evaluating the accumulation of errors arising in surveys due to the design, collection, processing and analysis of statistical data (Biemer 2010). TSE is a general data quality framework and can be applied on any type of data collection (Lyberg 2012), including JavaScript tracking.

This thesis is a thinkpiece where the author strives to emphasize the importance of high data quality in Big Data by applying an established data quality framework (the Total Survey Error pardigm) to a special type of data collection (JavaScript tracking) that can be incorporated into a Big Data context.

The first section of this thesis will establish the outset for this work by introducing the concept of Big Data and the field of web analytics. This section further establishes what is meant by 'online behavior', and then introduce the landscape of methods available for

collecting data on online behavior. JavaScript tracking, a prominent on-site method for collecting quantitative online behavior data, will then be introduced in detail, followed by an introduction of the statistical data quality framework Total Survey Error as a subset of the more general quality framework Total Survey Quality.

The second section of the thesis will look at five types of nonsampling error (specification error, frame error, nonresponse error, measurement error and data processing error) and elaborate on particular cases where nonsampling errors occur in client side on-site tracking. Methods and approaches for handling these nonsampling errors will be suggested and evaluated.

In the third section the author will present his conclusions and put forward suggestions on future topics that are related to quality of Big Data in general and data quality in online behavior tracking in particular.

# 1. Background

## 1.1. Big Data, the growth of the Internet and the web analytics industry

Big Data is recently featured in special issues of major statistical and economic magazines (Champkin 2012a, Micklethwait 2010) and emphasized during keynote presentations of major conferences (Rencher 2013). Big Data revolves around collecting, managing and working with Big Data quantities, and ensuring that one has the adequate processes and tools for doing so (Keller et al 2012).

Laney (2001) defines Big Data through three dimensions of data: Volume, variety and velocity. Big *volumes* of data are collected from financial transaction systems (Palmer 2013, Halevi and Moed 2012), online social media streams (Lansdall-Welfare et al 2012, Champkin 2012b) and the Hubble Telescope (Feigelson and Babu 2012). There is a great *variety* of data sources, ranging from peoples political preferences to their commercial information to their health information (Keller et al 2012, Mayer Schonberger and Cukier 2013). Some of these data sources are structured, but many are unstructured. Only a small amount of the data is collected with a purpose. The sheer majority of the data is so called organic data, data collected without any specific research question in mind, but stored just because it is possible to store it (Groves 2011). The third dimension is data *velocity*. When working with big volumes of data high speed systems and software are crucial in order to handle the data quantities (Hilberg 2012). High velocity is also important because of the increasing demand of real time analysis, a trend that is pushed by customers demand for faster service and faster information (Rencher 2013, Lorentz 2013).

Since late 2012 IBM has been introducing a fourth dimension to Big Data: Data *veracity* (Mastrangelo 2012). According to IBM 1 in 3 business leaders don't trust the data they base their decisions on (IBM 2013). This is alarming and calls for action. Trust in data is based on the accuracy of the data, and it is emphasized that all types of data sources need quality control procedures.

The Internet has grown very fast in the last couple of years. More people have access to the Internet, new websites are continuously created, and Internet is accessible on portable devices like SmartPhones (like the Apple iPhone series, Samsung Galaxy series and various HTC models) and tablets (like the Apple iPad, Amazon's Kindle Fire and Barnes & Noble's Nook). The number of registered website top domains have doubled from roughly 120 000 000 in 2006 to roughly 252 000 000 by the end of 2012, with a growth of 26,6 million domains between 2011 and 2012 (Internetstatistik.se 2013). In 2006 49% of all households in the EU27 had Internet access; in 2012 the coverage had increased to 76% (Eurostat 2013). In Sweden 77% had Internet access in 2006; this had increased to 92% in 2012 (Eurostat 2013). The worldwide number of mobile Internet subscribers using SmartPhones and Tablets has increased from 268 000 in 2007 to approximately 1 186 000 in 2011 (Internetstatistik.se 2012).

As the Internet has been growing, so has the number of companies that base some (or all) of their business online, and the trend is for continued growth of online business in both Europe and the US (Gill and Wigder 2013, Mulpuru 2011). Retailers and other companies doing business online realize that if they gain an understanding of how people behave on their websites they will know more about how to change and develop their sites in order to enhance the experiences for their visitors and customers (Kaushik 2009). The web analytics industry is dedicated towards interpreting and understanding peoples' online behavior, and help companies and organizations manage business development based on these understandings (Kaushik 2009). Analyzing peoples' online behavior in and of itself is good, but even more can be gained by incorporating web analytics into a greater Big Data spectrum (Rencher 2013).

## 1.2. Online behavior

How people use the Internet, or rather how they *behave online*, is very individual. Some people use it mainly for communication, for example through the use of online communities (like Facebook and Twitter). Others use it mainly for sharing or gathering information, for example by reading newssites (like New York Times and Dagens Nyheter) and managing webblogs (like Paul Krugmans blog on New York Times website and the Statistics is Beautiful blog]. Others use it to buy products from Internet retailers, for example clothes (like H&M and Bik Bok) and media (like Amazon and CDON). Yet others use the Internet mainly for entertainment, for example by visiting video streaming websites (like YouTube or Netflix) or gaming websites (like Flash-spel.se and Pokerstars.se).

In this thesis people that access the Internet will be referred to as users. Users will be referred to as visitors when they visit websites. It is called a session when a user accesses the internet. A session starts when the person accesses a web page on a website, and ends when the person stops accessing the website. During a session the person *move around* and they *interact* with web pages on various websites. The way people move around generates a web page movement pattern: What page they go to when they are done with the page that they are currently visiting, and for how long they stayed on that particular page. An example of a movement pattern would be that a person opens up a web browser and types www.newyorktimes.com. When he comes to the front page of the newspaper site he stays on that page for 14 seconds, then he clicks on a link to an article. He stays on the article page for 34 seconds, and then types www.youtube.com. He stays on the front page of the video site for 7 seconds before he makes a search for a video and comes to a search result page where he stays for 9 seconds before clicking on one of the videos.

Here is a summary of this pattern:

| | | |
|---|---|---|
| page 1 | New York Times front page | 14 seconds |
| page 2 | New York Times article page | 34 seconds |
| page 3 | YouTube front page | 7 seconds |
| page 4 | YouTube search results page | 9 seconds |

What people *do* while they are visiting a webpage can be viewed as how they interact with the page. Continuing with the same example as above, the person might not have interacted with the first page except for reading the headlines of the articles. While on the second page, he might have interacted with the page by scrolling down the page to read the whole article. On the third page he interacted with the page by clicking the search field and typing his search, for example 'Make Good Art Neil Gaiman'. On the fourth page he might have scrolled up and down among the results before deciding on with video to click and view.

| **Example movement:** | **Example interaction:** |
|---|---|
| Previous page | Clicking on buttons and textfields |
| Time on page | Typing in textfields |
| Next page | Scrolling up and down on the page |

*Figure 1. Summary of example behaviors*

### 1.2.1. Needs of information from online behavior data

By formalizing goals for a website and collecting data on their visitors' behavior a company or an organization can evaluate if their visitors are behaving as the company wants them to do. For example, a company selling products online wants their visitors to look at, and buy, their products. By collecting movement data on how many of their visitors that go to the shopping card inventory page and then to the checkout page and then leave the website (without completing the purchase) they can get an indication whether their checkout page looks good and is approachable for potential customers, or if it 'scares' potential customers away from the website.

There are different kinds of stakeholders, and they have different information needs. Accountable stakeholders are the stakeholders that are ultimately responsible that a task is completed and adequately delivered (Smith and Erwin 2007). An example of an accountable stakeholder is the company CEO. Example information needs for the company CEO is product group sales comparisons and marketing campaign evaluations.

Responsible stakeholders are actively responsible for achieving tasks (Smith and Erwin 2007). Examples of responsible stakeholders are marketing managers and website content managers. Example information needs for marketing managers is data regarding sales that can be connected to different marketing campaigns, and use the data to determine which campaigns proved successful and which did not. Example information needs for content

managers is data on which website content the visitors are interacting with and to what degree they interact with the content, for example if the visitors are more prone to watch a video if it is featured in the top right position of the page instead of in the top left position of the page.

Consulted stakeholders are subject matter experts who are consulted during the planning phase of a project. Consulted stakeholders are said to be 'in the loop' and often partake in active discussion about how progress and direction of a project. Informed stakeholders are individuals with particular interest in the project that should be informed once the project has been initiated (Smith and Erwin 2007).

### 1.2.2. The online behavior data collection landscape

There are both qualitative and quantitative methods for collecting data on peoples' online behavior. Some of the more common qualitative methods are discussion forums, usability testing, card sorting and qualitative interviews. Some of the more common quantitative methods are surveys and various types of on-site tracking. Figure 2 show a map of different methods.

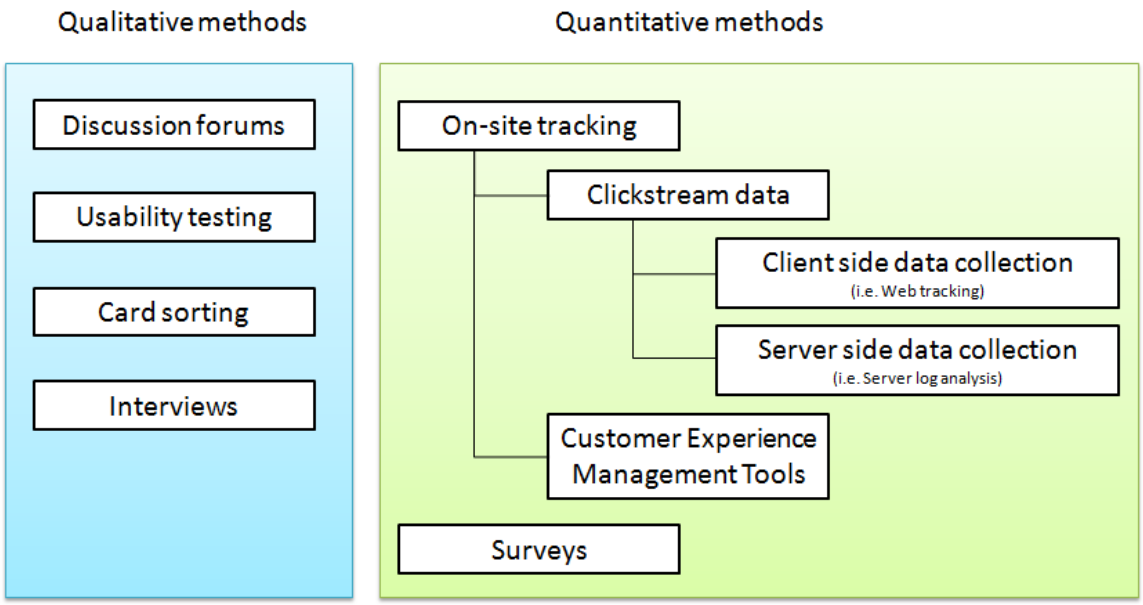


*Figure 2. Map of different online behavior data collection methods*

Each of these methods and approaches has their strengths and weaknesses, and preferred situations when to be used. Each method will be presented and commented on below.

- *Qualitative methods*

*Discussion forums*, commonly referred to as focus groups, feature a group of people (of which one is the group- and discussion leader) that sit down and discuss various topics of the group leaders choice. The setting is interactive and the participants are allowed to comment and elaborate on each others opinions. Meetings can be conducted live, for example at a company or at a café, or in a virtual space, for example in a chat-room online (Creswell 2012).

In *usability testing* a respondent is presented with a design (for example on a website) and a scenario to carry out (for example to search for, and buy a product). Data is collected on the respondents behavior, data which is later compared to predetermined checkpoints and goals set up by the tester. A design is deemed successful if the respondent manages to reach the checkpoints and achieve the goals without any hardships. Respondent performance is often graded using a 'usability scale', and different scenarios are compared to each other through this scale. Background data on the respondent is collected as well as data on whether the respondent achieved the predetermined goals or not, for example if the person clicked on the intended icon on the page or not. Other kinds of session data can also be collected, for example eye movements, click-patterns, time spent on particular pages, etc (Rubin and Chisnell 2008).

When constructing a page design it is possible to use *card sorting* as method for gathering respondent opinions on the design. 'Cards' are printed out containing different parts of the design. These cards are then presented to a respondent unsorted, and the respondent is instructed to sort them in a design of his or her liking. A short interview can take place before, during, or after the respondent has made his or her design (Spencer 2009).

*Qualitative interviews* can be performed about the respondents' attitudes towards existing website designs or planned future designs. The formality of the interview depends on the occasion and the interviewer structure can range from very open (the interviews is more or less a discussion about a subject chosen by the interviewer) to very structural (reading a questionnaire word by word). Interviews performed are often a mix of these two extreme cases (Ryen 2004, Creswell 2012).

- *Quantitative methods*

Ordinary *surveys* can be designed to explore respondents' attitudes and behaviors online. Surveys can either be probability based (for examle mail surveys or email surveys) or non-probabilistic (for example pop-up questionnaires or polls featured on web pages). Questions about how the respondent behaves online can be asked, but also questions about why the respondent behaves the way they do (Couper 2000).

*On-site tracking* tracks how the visitors move around and how they behave on the website. On-site tracking data collection methods that are coded into the source code of the website they are designed to collect data for (this is commonly referred to as "code is implemented" on the pages). On-site tracking is site-bound, which means that it only collects data for the particular website that it have been designed and implemented for (Schaefer et al. 2012, Google 2013, Omniture 2010).

*Customer experience management* tools works more or less like a recording of the events taking place during a visitors' session. It records everything that happens on the monitor for the visitor. The method was originally designed to fight forging, but is also widely used in customer support. Data collected through customer experience management tools have a very high level of detail, but are harder to aggregate than other quantitative methods (Schmitt 2003).

A variation of on-site tracking is *clickstream* data collection. Clickstream data can vaguely be described as a stream of clicks, i.e. data about where visitors' click on web pages and in what order they click. Clicking is either an interaction with the page or a movement action. There are two major technical solutions for clickstream data collection: server side tracking and client side tracking. Both types of methods are used to collect predetermined data, i.e. they collect data for clicks and actions that have been specified in the source code of the website (ZOHO Corporation 2012, Omniture 2010).

*Server side tracking* sets up the server to collect data when visitors interact with and move around on the pages of the website. Server side tracking is limited to collecting data on things that the server is aware of, for example what content has been requested and how many times it has been requested (Schaefer et al. 2012, ZOHO Corporation 2012).

*Client side tracking* is set up so that when the visitors interact with and move around on the website their browsers will send data to a database. Client side tracking can collect data on things that the server is aware of (like server side tracking) and can also collect data about things that the user knows but the server is unaware of, for example what button the user clicked on the page or what the visitor wrote in some text field on the page (Omniture 2010).

Client side on-site tracking has the capability of collecting big amounts of data and is a prominent and widely used method for collecting quantitative data on peoples' online behavior. One reason why client side on-site tracking is popular for collecting online behavior data is because the data collection tool providers not only offer to store the data that is collected for the website owners, but also provide a data access tool that the website owners can use to access their data (Google 2013, Omniture 2010).

## 1.3. Collecting data on online behavior

In order to understand the statistical errors related to client side on-site data collection, one must first understand the technical aspects of the data collection process.

### 1.3.1. How client side on-site clickstream data collection works technically

Websites consist of web pages. Websites and their web pages are accessed through web browsers. These browsers display web pages on monitors, for example a computer screen or the screen of a portable device (for example a tablet or a SmartPhone). What is displayed on the monitor is a representation of the web pages' source code, i.e. everything that is visible on the monitor has been written in the source code (Musciano and Kennedy 2006). Figure 3 show an example of how source code can look for a website.

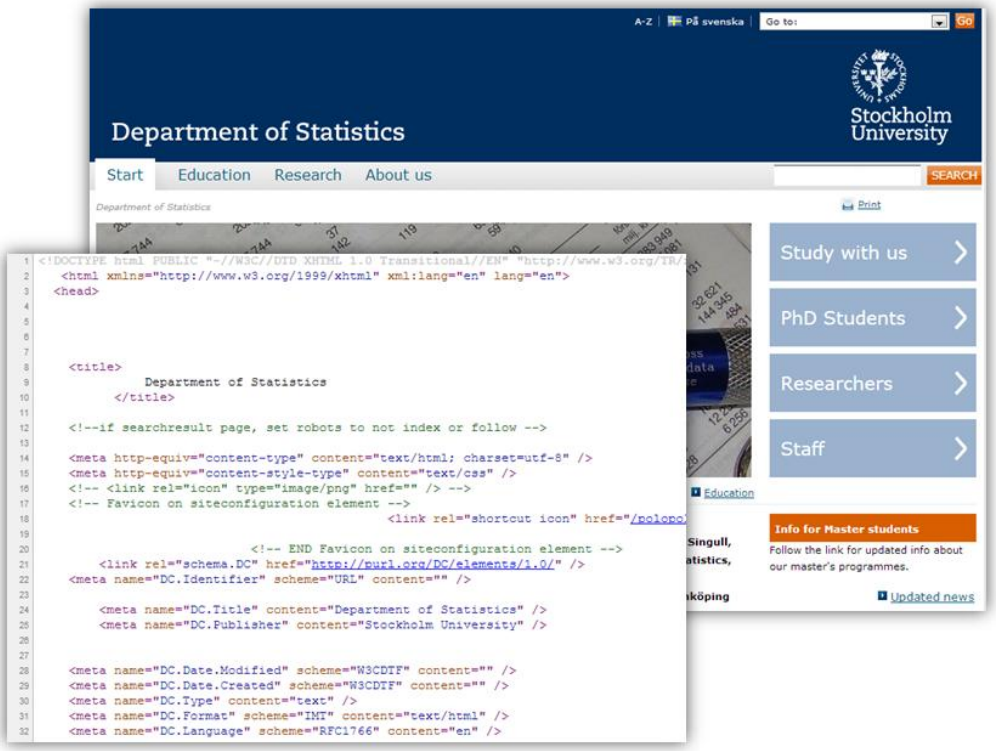


*Figure 3. Example website and an excerpt of its source code.*

The source code is usually written using HTML (HyperText Markup Language). HTML provides hypertext content that is rendered on the monitor as well as links that will redirect the visitor to another web page. Examples of HTML hypertext content are style sheets, images and scripts. Style sheets enhance how a web page looks on the monitor, what colors are used and in what font text will be presented. Images are the pictures displayed on the monitor. Scripts are small predetermined actions that can be executed when a criterion is met (Musciano and Kennedy 2006). Figure 4 show some examples of different tags that are used to write HTML documents.

Style sheet tag example:

```
<html>
<head>
<style type="text/css">
h1 {color:red;}
p {color:blue;}
</style>
</head>

<body>
<h1>A heading</h1>
<p>A paragraph.</p>
</body>
</html>
```

Image tag example:

```
<img src="smiley.gif" alt="Smiley face" height="42" width="42">
```

Script tag example:

```
<script>
document.write("Hello World!")
</script>
```

*Figure 4. Examples of style sheet tags, an image tag and a script tag.*

When a visitor comes to a web page, a request for that page is sent to the server storing the HTML documents for the page. When the server sends back the HTML document, the web browser starts to render the web page by reading the tags in the HTML file from top to bottom (Gourley et al 2002). This rendering process is commonly referred to as "the

page is loading". Tags either create content (style sheets tags, for example) or create sub-requests for content located on the server (image tags, for example) (Musciano and Kennedy 2006). Whenever content on the server is requested we say that "a server call is made".

A <script> can be set up to make a request for a small picture with the size of 1x1 pixels. This picture is commonly referred to as a "tracking pixel". The browser can *send* information to the server within the pixel request, and the information that is sent with the server call for the pixel can be stored in a database on the server. Both static and dynamic information can be sent with the pixel request. Static data is data that is equal for all visitors (for example what time the server call was made) and dynamic data is data that is unique for each visitor (for example which web browser he or she is using). This data collection process is called client side data collection because it is the clients' web browser that sends data to the data base (Flanagan 2011).

Figure 5 show an example rendering process for the made up website www.example.org. First the users' browser send a request for the HTML file (and the server sends the file to the browser to be read), then the browser starts to render the web page. When the browser encounters a <img> tag it sends a request for that image to the server (and the server sends the image to the browser to be rendered), then the browser continues to render the page. Then the browser encounters a <script> that requests a tracking pixel, and together with the request for the tracking pixel it also send behavior data to a database.

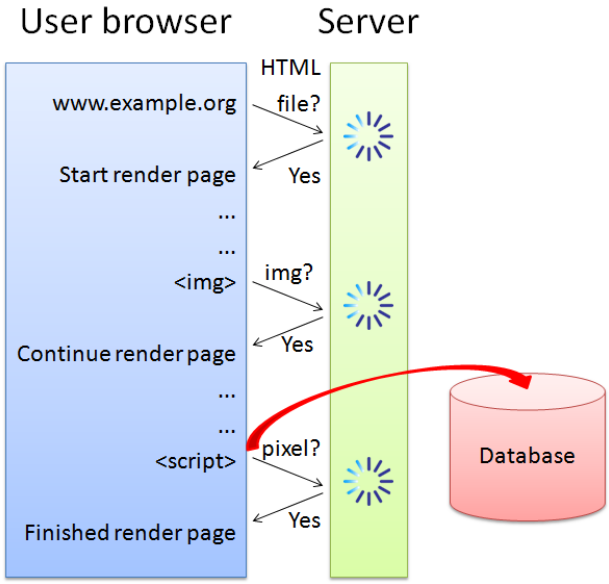


*Figure 5. Example rendering process including a tracking pixel request that send data to a database.*

<script> that makes pixel server calls are often programmed using JavaScript. JavaScript is a programming language that enables the tracking pixel server calls to be made *after* the web page has loaded completely and rendered all other contents on the page (Flanagan 2011). By having JavaScript making the pixel request after the page has loaded successfully, the data collection will not interfere with the users' web browsing

experience. JavaScript also enables that server calls can be made without the page being rendered. These triggered server calls are useful when data should be collected on some kind of online behavior that is not related to a page load, for example if the visitor clicks or starts watching a piece of film on the website (Flanagan 2011). JavaScript is the most common way to code <script> for collecting data on online behavior because it is the scripting language most web browsers support (Flanagan 2011).

Scripts are executed by the users' browser, and it is possible for the user to install software that blocks scripts and interrupts the <script> from being executed (for example the NoScript Security Suite addon for Mozilla FireFox or by activating Disable Script under Security in Internet Options in the Internet Explorer browser). By using the <noscript> tag programmers can code alternate contents that will be displayed or initiated if <script> is blocked from being executed (Musciano and Kennedy 2006).

When a visitor comes to a website, a small text file called a "cookie" can be stored in his or her web browser. These cookies are either set by the website (these cookies are called first party cookies) or by external scripts (these cookies are called third party cookies) (Gourley et al 2002). Depending on the technical solution, first party cookies or third party cookies are used for collecting online behavior data.

Cookies can store any information about the user. This information can be retrieved by the browser later when it is used to visit the same website again (Gourley et al 2002). For example, a user can let the website store his or her password in the cookie so that the next time he or she comes to the page they will not have to type the password again. Instead it will already have been filled with the password retrieved by the site from the cookie. In client side on-site data collection, cookies are used to recognize returning visitors to the site. Cookies can also store many types of data, which can be retrieved and sent together with behavior data in the tracking pixel server call (Omniture 2010).

As a clarification of the naming structure, *JavaScript tracking* is a specific form of *client side on-site online behavior clickstream data collection*, and;

- JavaScript is a programming language for writing website source code.
- Tracking implies that data is collected on the websites visitors' behavior without them being specifically notified about it taking place.
- Clickstream data is the type of online behavior data that is collected with client side on-site tracking.
- On-site tracking implies that the data collection method is site-bound and will only collect data on visitors' behavior to that particular website.
- Client side data collection implies that the website source code has the visitors' browser send the data that is collected to the database for storage (instead of having the web server send the data to the database, as is the case for server side data collection).

### 1.3.2. What data can be collected with client side on-site data collection?

There are three types of data that can be collected about a visitor. *Movement pattern data* is all data regarding how the visitor came to and moves around on the website. *Interaction data* is all data regarding how the visitor interacts with the contents on the web pages he or she is visiting. *Metadata* is data about the visitor and their properties. Each type of data is included and collected by the server for each server call (Omniture 2010). For example, if a visitor to an online clothing store comes to a product page for a pair of jeans, movement pattern data could be collected indicating that the visitor came to this product page from a search results page, that the visitor spent 19 seconds on the search results page before clicking to come to the page for the jeans, and that the denoted 'page name' of the jeans product page is 'BlackBaggyJeansMaleFit'.

Interaction data that might be collected could be, for instance, that the visitor clicked on the 'more info' button on the previous page before clicking to go to the product page. Metadata that could be collected could be, for instance, that the visitor was a returning visitor to the website (information stored in a cookie), that during the previous visit the visitor had visited jeans product pages and shirt product pages, that during this visit the visitor has visited jeans product pages, that the visitor is using the web browser Google Chrome and that the visitor is using a 1280x720 resolution on his or her monitor while visiting the product page.

There are different types of movement pattern data, interaction data and metadata. Figure 6 visualizes the different types and groups of data that can be collected with JavaScript tracking.

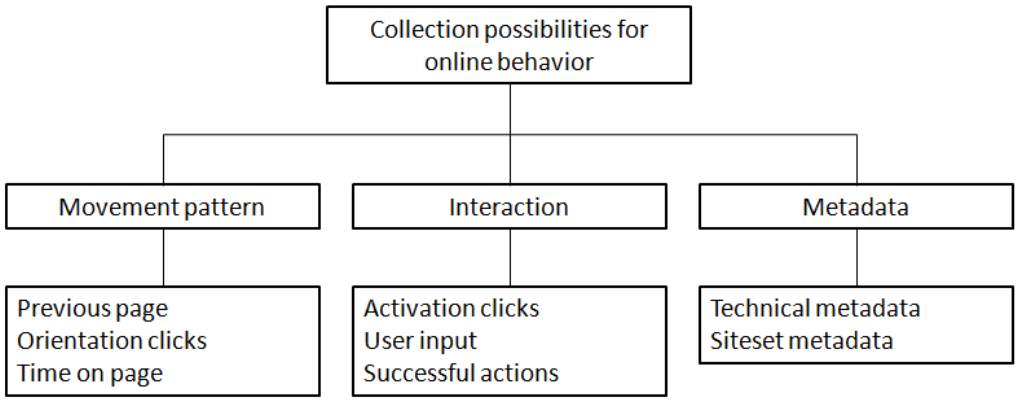


*Figure 6. Three types of data that can be collected for peoples' online behavior and their respective sub groups*

Examples of movement pattern data include:

- Previous page – When a visitor comes to a web page, data can be collected for which page he or she came from. This is limited to pages within the website, however, and it is not possible to see if the visitor came from another website.
- Orientation clicks – When a visitor comes to a page, data can be collected on which link (i.e. the 'name' on the link) the visitor clicked on to get to the new page.
- Time on page – When a visitor comes to a page, data can be collected on for how long the visitor stayed on the previous page.

Examples of interaction data include:

- Activation clicks – Data can be collected when a visitor clicks on a button (for example to start a video or to show more information about a product) or interacts with the page in any way (for example scrolling down the screen or starting to write in a text field).
- User input – Data can be collected on what a visitor types in a text field (for example filling in an open text field for performing a search on the website).
- Successful action – Data can be collected whenever a predetermined action is performed by a visitor (for example if a visitor applies for a loan on a bank site after visiting the loan information page).

Examples of metadata include:

- Technical metadata – Data can be collected on technical properties that the visitor is using (for example what brand of web browser or what version of JavaScript he or she is using).
- Siteset metadata – Data can be collected on predetermined visitor status (for example determining if the visitor is a new visitor or a returning visitor or through what marketing channel the visitor came to the page).

All data that is collected through JavaScript tracking is time stamped (Omniture 2010). As data collection is continuous, time intervals have to be constructed by the person analyzing the data in order to enable analysis. Constructing equally sized time intervals and relating these intervals to each other enable over-time comparisons. Because of this, over-time comparisons are very flexible.

## 1.4. Data quality frameworks

In order to enable data to drive business development, that data must be of high quality. There are many concepts and definitions of data quality, but one commonly used in survey methodology is *fitness for use* (Biemer 2010), first proposed by Juran in the 1940's (Juran and Gryna 1988). The fitness for use concept recognizes that producers of data and users of data (not to be mixed with an Internet user) often perceive the concept

of high quality quite differently. Users in traditional survey methodology can be both accountable stakeholders and responsible stakeholders, i.e. sometimes they have overall responsibility (for example the company CEO) and sometimes they are directly responsible for the data collection process (for example a marketing manager).

A commonly used framework for visualizing fitness for use is the Total Survey Quality (TSQ) framework (Biemer 2010). Even though TSQ has been developed for evaluating survey process quality and survey data quality, the framework is applicable for any type of data collection (Biemer 2010, Lyberg 2012). The applied TSQ frameworks vary slightly depending on the organization for which they are implemented, but most frameworks contain versions of the nine dimensions presented in table 1. The descriptions are adjusted to fit for an organization working with online behavior data collection.

| **Dimension** | **Description** |
|---|---|
| Accuracy | Estimates produced are valid for what they are supposed to measure |
| Credibility | Data are considered trustworthy by the web analytics community |
| Comparability | Demographic, spatial and over time comparisons are valid |
| Usability / Interpretability | Documentation is clear and process metadata are well-managed |
| Relevance | Data are relevant and satisfies organizational needs |
| Accessibility | Data is easily accessible for the users of data |
| Timeliness / Punctuality | Data is available on time and in a timely fashion for the end users |
| Completeness | Data is rich enough to satisfy analysis objectives |
| Coherence | Estimates from different source can be reliably combined |

*Table 1. Total Survey Quality dimensions adjusted for online behavior data collection. Based on Table 1 in TSE: Design, Implementation and Evaluation (Biemer 2010)*

Out of the nine dimensions, accuracy corresponds to the actual data quality. Total Survey Error (TSE) is a framework that identifies and evaluates errors accumulated during the design, collection, processing and analysis of survey data (Biemer 2010). In survey methodology TSE is separated into sampling errors and nonsampling errors. Sampling errors are errors that occur due to the sampling scheme, the sample size and the choice of the estimators for the survey. Nonsampling errors occur either due to erroneous data collection specifications, frame construction issues, survey nonresponses, measurement issues or data processing issues (Biemer 2010). These error sources are formulated to fit a survey situation, yet the nature of the errors should be the same for all data collection processes. Even though some naming conventions will have to be shoehorned into place, the error sources are real and applicable for online behavior data collection as well as ordinary survey data collection (Biemer 2010, Lyberg 2012). Figure 7 displays TSE components and their relationship to TSQ.

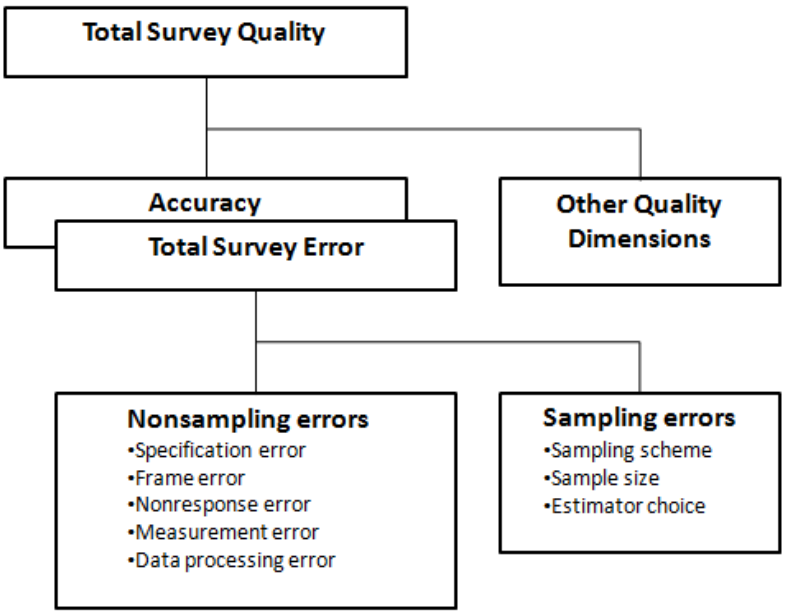


*Figure 7. Total Survey Error and its components as a part of Total Survey Quality*

Errors in data jeopardize inference by diminishing the accuracy of the estimates produced from the data. Accurate estimates have a small variance and a small bias, i.e. a small TSE. TSE is estimated by the Mean Squared Error, which is defined as the squared bias of the estimate added to the variance of the estimate (i.e., MSE = Bias$^2$ + Variance). Bias comes from systematic variation in the sample, while variance comes from variable variation in the sample. While variance affects the accuracy of the estimator, bias skews and influences the estimator. As such it is desirable for the data collection designer to get rid of bias (Biemer 2010).

According to the *total survey error paradigm*, major sources of error should be identified and have resources allocated to reduce their effect on the estimates (Biemer 2010). This thesis will focus on identifying the nonsampling errors for client side on-site online behavior data collection rather than estimating them. The identified errors will be categorized as one of the five nonsampling error types. No effort will be put into estimating the error sources, but comments will be put forward on the authors' ad-hoc estimated severity of the errors. Further, the author will produce recommendations for how to address or approach the identified errors.

# 2. Evaluating online behavior data quality

Companies and organizations use client side on-site clickstream data collection (commonly referred to as web tracking) to collect data on how their visitors move around and interact with their website. This behavior data is analyzing in itself or is connected with other data sources to form a Big Data network that is used to aid business development.

For data to serve this purpose is has to be of high quality, however, and that no bias is imposed on the data. Biased data introduces the risk of drawing erroneous conclusion, and can lead do decisions that will cost the company or organization a lot of money. Therefore it is important to identify error sources and estimate bias. It is important to not lose sight of the underlying details when faced with the sheer possibilities that Big Data offer.

According to the Total Survey Error framework, errors can be separated into sampling errors and nonsampling errors. Web tracking methodology puts these error sources in a different context than ordinary survey methodology. In ordinary survey work it is expensive to increase the sample size, but the situation is very different for data collection using web tracking. Some tracking solution providers charge a small fee for every tracking pixel server call (for example SiteCatalyst by Adobe), while other tracking solution providers offer the data collection service for free (for example Google Analytics by Google). As data collection is relatively cheap, sampling is rarely done in client side on-site data collection. For this reason, sampling errors will not be addressed in this thesis.

Nonsampling error can be separated into five different types of errors: Specification errors, frame errors, nonresponse errors, measurement errors and data processing errors.

In a web tracking context specification error is about establishing good channels of communication between different actors working with online behavior data collection. Specification error also concerns the fundamental flaw of on-site tracking; that the data collection depends on visitors' actual interaction with their browsers for data to be collected. Frame error focus on the troublesome many-to-one relationship between people and the number of web browsers and devices they are using, as well as the identification problem of who or whom are actually using a browser to visit a website.

Nonresponse errors stem from users deciding to block important features that enable client side on-site tracking to function properly. Measurement errors cover errors that occur as byproducts of certain user behaviors that the tracking solution is not able to handle. Data processing errors are errors that occur if people are not careful when handling and working with collected data.

Figure 8 display an overview of all nonsampling error sources that will be discussed in this thesis, as well as the error type that they correspond to.

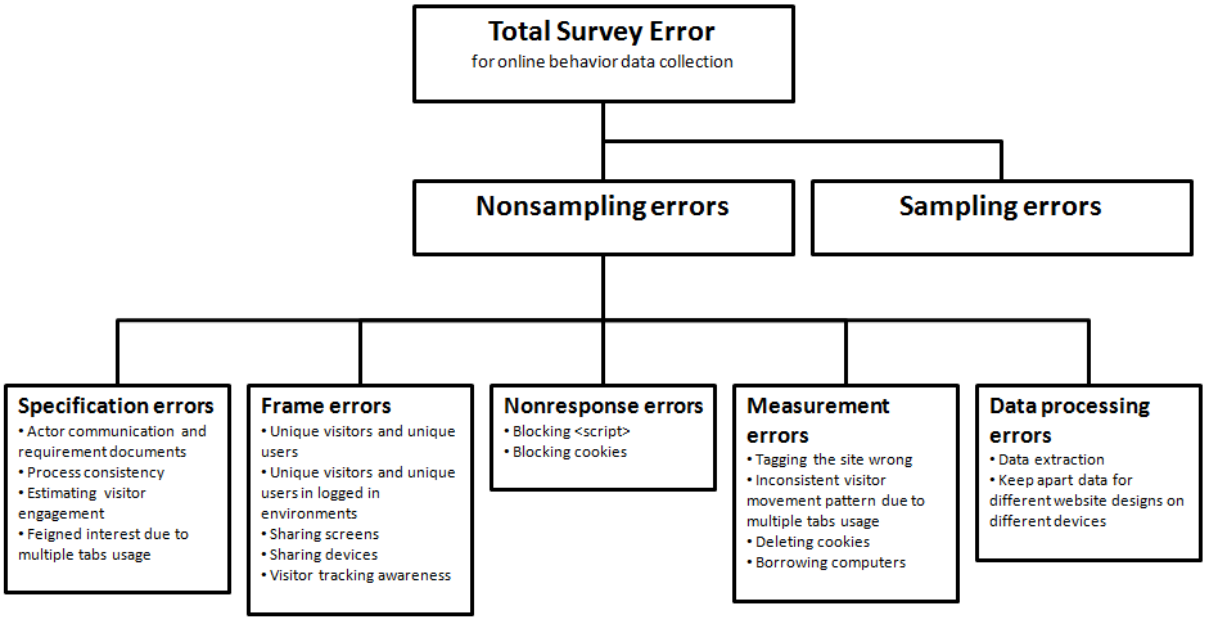


*Figure 8. Nonsampling error and its components as a part of Total Survey Error*

Section 2.1 will cover specification errors that occur in online behavior data collection. Section 2.2 will elaborate on frame errors, while section 2.3 will address nonresponse errors sources. Section 2.4 will identify and discuss measurement errors, and finally section 2.5 will take a closer look at data processing error sources.

## 2.1. Specification error

In ordinary survey work the specification error is the mismatch between the research question and the survey data (Biemer 2010). In a web tracking context the specification error is the difference between what the end user wants and what he or she gets. This error often has roots in miscommunication between the different actors involved in the data collection project. In ordinary survey work the most common actors are the survey sponsors, researchers, questionnaire designers and data analyst (Biemer 2010). In client side on-site data collection the most common actors are the responsible stakeholders, data analysts and programmers.

Responsible stakeholders work operationally and want to collect data that the organization or company will be able to base business decisions on. Usually these stakeholders have no training in exactly what data can be collected in a given technical environment, nor do they have training in how to analyze online behavior data. Instead they have a good understanding of business development and fitting the data collection and analysis process into a larger perspective. Responsible stakeholders are often the end users of data collected and thus require good specifications in order to guarantee that the collected data fulfills the organizations' needs. Example stakeholders in client side on-site

data collection are online marketing managers, website content managers and e-commerce managers.

Programmers have training in what types of data can be collected in a given technical environment and what data cannot. Usually programmers focus more on technical aspects of data collection; code has to be properly written, server calls should be sent as intended and data should come through to the databases without malfunctions. Programmers require good specifications in order to assure that the intended data is being collected properly.

Data analysts have analytical training and are experienced in working with data extraction tools and data reporting tools. They produce, analyze, and deliver data to stakeholders. They also perform quality control of the data that is collected to the databases, and are expected to discuss with the programmer if there is a problem with the data or the data collection. Data analysts are often required to have a holistic attitude and keep a good dialogue with stakeholders to get a good understanding of the stakeholders' long term plans and needs, and with programmers to understand the limitations of the current technical environment.

An example of an online behavior data collection setup process could be that an online marketing manager contacts the analyst and informs him or her of a need for data. The analyst decides what data would be suitable for the marketing manager and writes data collection requirements for the programmer to create and implement. Data collection starts after the programmer has written and implemented the tracking code. When a satisfactory amount of data has been collected the analyst will either present the data for the marketing manager to analyze for himself/herself, or the analyst can analyze it and produce the results for the marketing manager. The manager can then work with the data to manage a potential business change.

Big Data discern between designed data and organic data (Groves 2011). Designed data is data that is collected with a purpose and a research in mind, while organic data is data that is collected and stored just because the data is generated and there is a possibility to store it. In traditional survey methodology specification error concerns only designed data collection. Web tracking methodology works a bit differently, because once the web tracking has been designed and implemented then data will be continuously collected until it is decided to remove the implemented code.

Picture the situation where web tracking is set up to collect data on a data flow that a stakeholder is responsible for. Designed data is collected. Then, several months later, the stakeholder no longer has a need for data from the data flow, and the analyst stops producing reports for the stakeholder. Data will still be collected, however, even though no reporting is done. The previously designed data has now changed into organic data.

Data might not have to be designed, but it is important to have a plan for its collection. Otherwise there is a risk for big data repositories with no clear responsible stakeholder or analyst. It is noteworthy that even though data storage is cheap in this day and age, there

is still a cost for storing data. These costs are not necessarily monetary. There are alternative costs like such as time and energy that have to be put into managing the data repositories and the creation of documentation of what type of data is stored in the repositories (or worse, the lack of documentation on what data is stored in the repositories). While organic data can be good and useful to keep around, it is important not to take data collection for granted and to always have a purpose in mind when establishing data collection processes.

Section 2.1.1 will discuss the importance of actor communication and the process around creating requirements document. Section 2.1.2 will talk about the importance of process consistency. Section 2.1.3 discusses the issues regarding estimating visitors' engagement towards a website, while section 2.1.4 looks into what errors that occur due to users tendency for using multiple tabs in their web browsers.

### 2.1.1. Actor communication and requirement documents

Requirement documents are documents that include instructions on what website source code that needs to be written and implemented on the website. The documents also specify what data will be collected through the code, and how the data should be used once it has been collected (Goldsmith 2004). Usually data analysts are responsible for the creation of the requirement document while being assisted by the responsible stakeholders (who establish the purpose of data collection) and the programmers (who establish the technical structure).

Even though analysts often find themselves with the task to produce the document, all actors share the responsibility of establishing requirement documents of high quality. Collaboration is crucial, and good communication between actors is very important. The collaboration process is improved if the different actors have at least a basic understanding of the other actors' competence and working roles. Two collaboration models can be set up from this, representing the two extremities of actor communication.

The first collaboration model represents the perfect scenario. In this scenario each actor has a good understanding of each of the other actors' roles, goals and limitations, and can communicate their opinions, thoughts and feelings in a manner that the other actors can relate to. An example of this would be that the both the responsible stakeholder and the data analyst know some basic programming and are accustomed to regular programming phrases and structure, and thus will have an easier time understanding the programmer when he or she expresses his or her opinions about some part of the data collection process. At the same time the programmer and the analyst would have a basic understanding of business development and the stakeholder and the programmer would have a basic understanding of statistical analysis and the tools that the data analyst is using. An illustration of the first model is presented in figure 9.

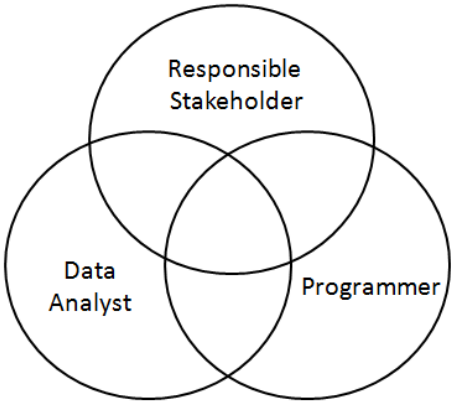


*Figure 9. Each actor has a basic understanding of the other actors' working fields*

The second collaboration model represents the nonperfect scenario. In this scenario it often falls on the data analyst to act as a communication link between responsible stakeholders and programmers. This collaboration model puts more communication responsibility on the analyst and less on the stakeholder and the programmer. An example of this would be that the stakeholder has no understanding of the programmers working environment and the programmer is not aware of what the data that his or her code produces will be used for, possibly the stakeholder and the programmer have never even met each other in person. Instead it is the analyst that communicates with the programmer so that useful data will be collected, and then reworks this data into a format that the responsible stakeholder can use in his or her work. An illustration of the second collaboration model is presented in figure 10.

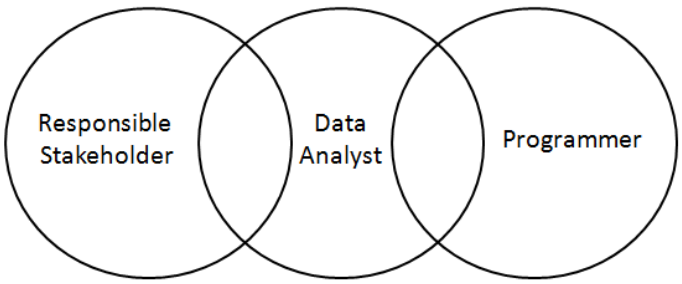


*Figure 10. The analyst acts as a link between stakeholder and programmer*

It is noteworthy that the two collaboration models are extreme representations of reality and that it is a progressive scale between the two. The first collaboration model is superior to the second model because it has the actors share the responsibility for requirement document creation, and enable them support and act at quality control for each other.

As it is a progressive scale, most companies and organizations will find themselves somewhere in the middle. The best way for organizations to transcend towards the superior model is through education. Organizations can conduct internal educational seminars and team building activities that help their employees to get to know each other and get a better understanding of each others' working conditions.

It is not unusual for every role to be occupied by several individuals in an organization, i.e. there are several stakeholders, several data analysts and several programmers that are working on the same data collection project. In these situations clear communication becomes even more important in order to create well structured requirement documents. Another common situation in organizations is when several roles are occupied by the same individual (for example the same person could be both data analyst and programmer). This situation requires less communication between roles, but put more responsibility on a single individual, which can be both good and bad.

### 2.1.2. Process consistency

A survey contacts its respondents and collects data during a particular time period, and responses to the survey are related back to the given time period. Web tracking is continuous: once the data collection code has been implemented on the site data will be collected continuously on the websites visitors' behavior (Omniture 2010). As data collection is continuous, administrating and updating data collection code becomes an iterative process.

Consistency is very important for iterative data collection processes. Without consistency every new addition to the design will have to be created from the bottom and up – and every time the wheel is reinvented there is a risk for standards to be breached and benchmarks to be overseen and ignored. Each of these faults introduces errors into the process as a whole.

Keeping good documentation of processes is one way of maintaining consistency. Documentation should be well structured, correct, and easy to read and understand for related parties (Hargis et al 2004). Documentation should be maintained for each role and actor involved with the online behavior data collection. Maintaining good documentation takes time and commitment, and there is a risk for it to have a low priority in day to day business because it is not part of the core business. It is important for managers within the organization to emphasize the importance of good documentation and to remind their workers of the importance of good documentation.

### 2.1.3. Estimating visitor engagement

Engagement is an unobservable variable when collecting data through client side on-site tracking. It is not possible to determine what a user is doing in front of his or her monitor in between interactions with the web pages (because the tracking solution only recognizes interactions that produce server calls). For example, if a visitor to a website stops moving around the site for a couple of minutes it is not possible to determine if the visitor is immersed with the site by reading a piece of text on the page, or if he or she has walked away from the monitor to get a cup of coffee.

*Time spent on site* is an established way of estimating visitor engagement in a website (Peterson 2006). Spending a long time on a webpage is seen as a higher level of engagement than spending a short time on the page. The average time spent on pages is usually calculated for separate pages, whole sections of the website, or for the site as a whole.

This estimate will consistently overestimate the visitor engagement. By deleting very small observations (for example if the visitor spent less than 2 seconds on the page) and very large observations (for example if the visitor spent more than 8 minutes on the page) the error in the estimate is decreased, because spending very little or very much time on a page indicates that the visitor was either just moving through or not engaged at all. In either case the time the visitor spent on the page will not serve to estimate the overall engagement towards the page and its contents, and should therefore be rid of. This is a simple method for reducing some error in the estimate, but the error source remains.

On a broader scale this error is more even severe than that it might seem at first, because it is a major limitation of on-site web tracking as a general method. By relying on visitors to interact with you through collection points on your website instead of, say, using a camera to directly observe everything they do, engagement becomes a truly unobservable variable. Hence, when trying to determine visitor engagement in some part of a website, another method of data collection could be suggested for more accurate behavior tracking. Examples of offline methods that could estimate engagement would be usability testing, focus groups, interviews or a survey. Customer experience management tools could also be used to record the visitors' behavior during a session. Most of the proposed methods are qualitative rather than quantitative, however, which stimulates overall understanding of single cases, but limits the opportunities for making inference.

### 2.1.4. Feigned interest due to multiple tabs usage

Modern web browsers have the option to have multiple tabs open and active at the same time. A tab is a browser window within the browser window. Using tabs enables the user to be on two different web pages at the same time. These pages can either be part of the same website or on different sites.

If the visitor is using two tabs to visit two different websites, then *time spent on site* will count upwards for the idle page while the user is browsing the website in the other tab, this results in feigned engagement for the page in the idle tab (and thus, overestimating the *time spent on site* estimate). For example, a user goes to page A on website M in tab 1 and stays there for 14 seconds. The user then opens tab 2 and goes to page X on website N, and stays there for 49 seconds. Then the user change back to tab 1 and interacts with page A for 12 seconds before clicking to go to page B (also on website M). The user stays on page B for 19 seconds. After that the user changes to tab 2 again and interacts with page X for 27 seconds before clicking to go to page Y (also on website N). An illustration of this example is presented in figure 11.

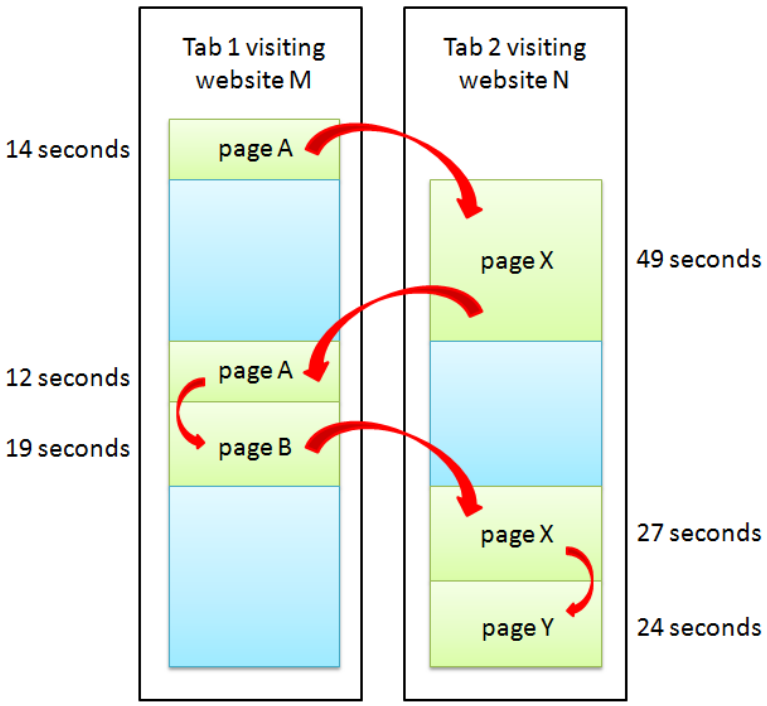


*Figure 11. Red arrows indicate the described visitors' movement pattern. Green boxes show how long the visitor is active on the page. Blue boxes show how much time the user is idle on the page.*

The overestimation of time spent on page A becomes clear as the user spent a total of 14 + 12 = 26 seconds on the page, but the tracking solution will have registered that the user spent 14 + 49 + 12 = 75 seconds on the page, because it also counts the time when the page is active but idle. Following the same logic the time spent on page X will be overestimated to 49 + 12 + 19 + 27 = 107 seconds instead of 49 + 27 = 76 seconds which is the real times spent on the page.

This error source might be solved technically by the web tracking solution providers. The solution providers could work to develop their tracking solutions so that they can identify if the user is active or idle in the tab, and only count *time spent on site* when the user is active in the tab and not when the user is idle.

Without technical development this error is harder to address. A survey could be conducted, a usability study could be set up, or discussion groups could be held in order to try and determine how people use tabs when browsing the Internet.

## 2.2. Frame error

In ordinary survey work frame errors stem from frame construction. For every survey there is a target population, and the intention with a frame is to produce a list where every individual in the list has a one-to-one correspondence with an individual in the desired target population. Frame errors are all errors that occur because the frame is not a perfect one-to-one fit with the target population. Not managing to establish a perfect fit can be because the elements in the target population are not present in the frame (undercoverage) or because there are elements present in the frame that are not part of the target population (overcoverage) (Biemer 2010).

Client side on-site online behavior data collection has no frame in any traditional manner. Instead of producing a list of names that is supposed to correspond well with the target population, web tracking aims to collect data on every browser that comes in contact with the website. This is a fundamental difference between surveys and web tracking. Frame errors for web tracking spawn from the relationship between people and the web browsers they are using, and how they are using them. In the same way that there is a desirable one-to-one relationship between the frame and the target population in an ordinary survey, it is desirable to have a one-to-one relationship between a person and a web browser in web tracking. This is typically not the case, however, as many people use several browsers on several devices (a many-to-one problem), and it is also common that people borrow each others' computers or share a web browser (an identification problem).

Seciton 2.2.1 will discuss the mismatch between users and how many devices they are using, and what errors that occurs due to this mismatch. Section 2.2.2 will look into how this mismatch is affected if the website has a logged-in environment for their visitors. Section 2.2.3 will discuss errors that occur when people share a screen and browse websites together, while section 2.2.4 will discuss errors that might occur when people share ownership of a computer. Section 2.2.5 will look at the bias that emerge when visitors to a site are not informed that data is being collected on their behavior.

### 2.2.1. Unique visitors and unique users

Client side on-site online behavior tracking tracks web browsers, and not individuals. Cookies that are stored onto browsers are the main tool for identifying returning visitors to a website. An individual is considered a user when he or she uses a web browser, and is considered a visitor when he or she uses a browser to visit a website. There is a many-to-one relationship between individuals and their web browsers, i.e. each new web browser on each new device that the individual is using will be interpreted as a new and unique visitor when data is collected, even though it is the same individual that is using multiple web browsers. The number of unique visitors that has come to a site is a common estimate of visitor frequency to the website (Peterson 2006). As long as there is not a one-to-one ratio between unique visitors and unique users, there will be an overestimation of the number of users that have visited the site.

Alternating between Mozilla Firefox and Google Chrome while browsing on your home computer is an example of changing the browser:device ratio to 2:1. One possible reason for this could be that the user experiences one browser to be faster at browsing text-based websites like Wikipedia, while the other browser is more proficient at playing videos on YouTube. Using several devices (but only using one type of browser), will skew the browser:device ratio to 1:2. An example of this could be that an individual uses Internet Explorer both on his or her work computer and on the home computer.

Figure 12 displays a 3x3 matrix of potential combinations that will all be identified as unique visitors from a web tracking perspective, but could all be the same unique user. A unique visitor is a user when accounting for a single browser on a single device only, while a unique user is a user for which all browsers and all devices have been accounted for. The number of unique visitor combinations that relate to a single user can be denoted *cookies per user*, because each single unique visitor combination is controlled for by a cookie stored in the web browser.

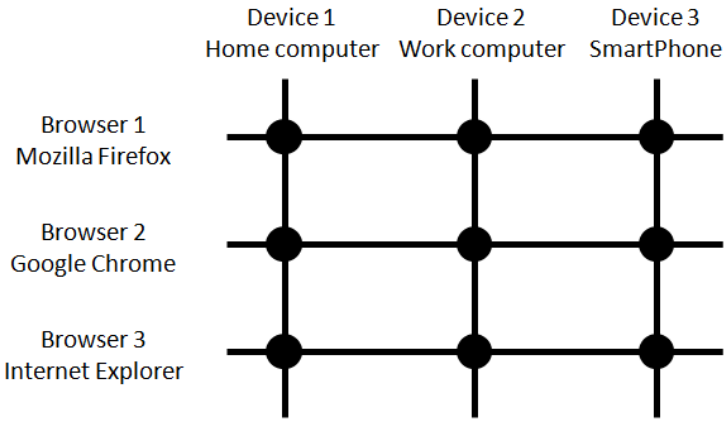


*Figure 12. Each dot marks a possible combination of devices and browsers, i.e. in the figure there are 9 possible browser/device combinations that could all be representing the same user.*

The situation in figure 12 displays the worst possible scenario. It is not very common that people use several browsers on the same device. It is more common, however, that people use multiple devices in their day to day life.

The bias generated by this error must be addressed in order to improve the quality of the estimate of visitor frequency. Some efforts have been made to estimate the number of cookies per user (Andersson 2012), but further research should be devoted to this issue. A survey could determine how many devices and browsers people are using, and also help distinguish if cookie per user ratio is different for different social groups. For example determine if teenagers have a higher cookie per user ratio than elderly people or business people.

### 2.2.2. Unique visitors and unique users in logged-in environments

Websites can provide the service for their visitors to create logins. A login is a website restricted alter ego that enables their visitor to do things and access content that would otherwise be unavailable to them, like reading restricted documents and articles, make purchases in the websites online store or comment on community forums. Logged-in visitors are a subset of all visitors to the site. All data that can be collected for non logged-in visitors can also be collected for logged-in visitors, but some data is unique for logged-in visitors and cannot be collected for non logged-in visitors.

For websites that utilize a logged-in environment it is possible to tie browsers and devices to unique users through their logged-in user profiles. The viability of this solution depends on the design of the website. There are three types of websites: Websites that have no logged-in environment (for example some university department websites), websites that have a logged-in environment but also has contents accessible for visitors that are not logged-in (for example e-commerce websites like Amazon.com), and websites that have a logged-in environment and only has content accessible to logged-in users (for example community websites like Facebook). Figure 13 illustrates these three types of websites.

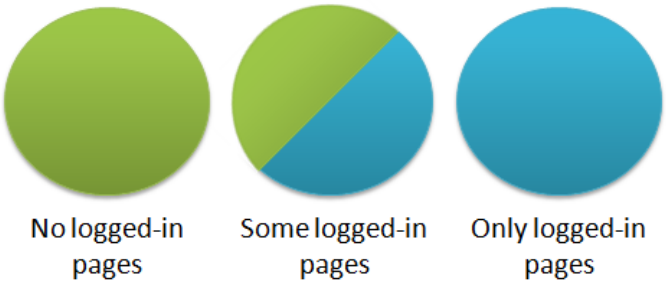


*Figure 13. Three types of websites.*

If the site is accessible only as a logged-in environment it is possible to use the logged-in profiles and correlate them to the number of visits from different devices and browsers, and determine the cookie per user ratio. This would have to be done continuously in order to maintain validity, but would also erase the bias that originates from a high cookies per user ratio. One has to remember, however, that the cookie per user ratio is unique for every website, so it is important to not draw too many conclusions from one websites findings to another website if there is not good cause for doing so.

If a website has some pages that are only accessible to logged-in visitors and some pages that are accessible to all visitors, then the logged-in feature will be inappropriate for estimating the cookies per user ratio for the site, because theoretically there should be a difference in behavior between visitors who come to the website for its logged-in features and visitors who come to the site for other features that do not require the visitor to log in.

### 2.2.3. Sharing screens

Client side on-site online behavior tracking collects data on web browsers, and it is not possible to determine what the user in front of the screen is doing while browsing. Neither is it possible to determine if the user is alone, or if there are several people sitting in front of the screen together. Two or more people could for example be sharing the screen because they are reading an information page together or planning a purchase together.

When two or more people are sharing the same screen the one-to-one frame individual representation assumption is violated. These two or more people represent the person that is usually using the browser that is being tracked, imposing bias on his or her usual behavior, as an individual might browse differently while browsing together with a friend or a relative than he or she would do when browsing alone.

This error source is quite severe, as peoples' online behavior changes quite drastically when they are browsing together with someone else. Data analysts require representative data in order to perform adequate analysis. They require visitors to behave like themselves, and not have their behavior influenced by other people.

At present there is no good method of controlling for shared screen behavior. However, this error source is possibly more common for certain kinds of websites that others. Websites that might not suffer so much from this error could be sites that are of a personal nature, like e-mail clients (like Hotmail and Gmail) and community websites (like Facebook and LinkedIn). Websites that might suffer more from this error could be sites that have a lot of information that many people may want to look at together, for example travel booking websites (like SuperSaverTravel and TravelPartner) and movie streaming services (like Netflix and Headweb). A survey could find out which websites are more prone to this shared screen behavior.

Analysts working with websites that suffer little from this behavior might be satisfied by knowing that the shared screen behavior makes up only a small part of their visitors. Analysts working with websites that suffer much from this behavior might either have to redefine their 'respondents' as 'one or more people' visiting their site, rather than 'one person', or invest into a method that would be able to find and distinguish shared screen behavior from ordinary single user behavior from among their visitors.

### 2.2.4. Sharing devices

Many households have computers that are shared by the members of the household. If the people that share a computer also share the same web browser, then several people will influence the behavior of the expected 'respondent' that the web browser represents. The same issue applies if cookies are not deleted between sessions for computers used on Internet cafés, then different people using the café computer will be considered the same individual when browsing the web.

As client side on-site online behavior tracking is site-bound, several criteria must be fulfilled for this error to be realized:

1. Two or more people need to share the same device, for example a stationary computer in a family home.
2. These people need to share the same user account on the device.
3. These people need to share the same web browser.
4. These people need to visit the same websites.

In Sweden and other developed countries (sometimes called minority countries) shared computers are not as common today as they were before. And with the mobile revolution (the explosion of mobile Internet platforms), Internet cafés are not as widely used for connecting to the Internet as before.

Yet, a survey could be conducted among households to try to determine what household constellations promote a shared device behavior, for example if it is more common for parents to share devices with their children than with each other, or if elderly spouses more commonly share devices than younger spouses.

### 2.2.5. Visitor tracking awareness

On-site tracking collect data on the websites visitors' behavior without the visitors being specifically notified about it taking place. If the visitors were aware that they are being tracked, there is a possibility that they would change their behavior while visiting the website. This implies that an awareness bias is imposed on the visitors' behavior when they get to know that they are being tracked.

The best way to approach this awareness bias is by informing the visitors that they are being tracked. How this information is phrased is very important, however, because visitors will react differently depending on how the message is communicated. For example aggressive phrasing like *'we collect data on your behavior'* or *'we are tracking you'* will probably be reacted to differently than softer phrasing like *'in order to improve your experience we are collecting some anonymous data about how you use our site'*.

How people react to the knowledge that they are being tracked will depend on which website they are visiting. For example people can be more reluctant towards government websites or other websites that store personal information about their visitors, but less

reluctant towards websites that has no information about them, like news websites and information websites like Wikipedia.

In 2009 the EU ePrivacy Directive from 2002 was updated so that website providers in Europe have to inform their visitors that they are being tracked when they visit their website. As of March 2012 there were still membership states that still had not transposed the directive into their national legislation (Rasmussen 2012), yet this is a good initiative towards making all visitors to websites aware that their behavior is being tracked. Even though the ePrivacy Directive was established for online users' privacy, the web analytics industry can free ride on this initiative in order get more high quality data.

## 2.3. Nonresponse error

In traditional survey methodology nonresponse occur when respondents do not respond to a questionnaire (either in full or partially). Nonresponse error occurs when the data that would be collected differs from the data that was collected, i.e. the parameter estimates will differ because there is a difference between respondents and nonrespondents (Biemer 2010).

It is called unit nonresponse when an individual neglects data to be collected entirely, and it is called item nonresponse when an individual neglect to provide parts of the data that is being asked for. It is called a break-off when a respondent starts to answer a question, but does not complete the answering process (Biemer 2010).

Nonresponse is either intentional or unintentional. In traditional survey methodology intentional nonresponse are refusals. Respondent chooses not to provide a particular piece of information to the data collector. This is for example common for sensitive questions. Unintentional nonresponse is when the respondent fails to provide information, for example by missing a question at the bottom of a page because he or she is going through a questionnaire too quickly or because the data collector was unable to get in contact with the respondent (Biemer 2010).

As client side on-site data collection code is incorporated into the website code there is no unintentional nonresponse for this data collection process. There is intentional nonresponse, however; users taking measures to assure that data is not collected on their behavior.

Section 2.3.1 will discuss errors that occur due to visitors' tendency to disable or block <script> from being initiated on websites. Section 2.3.2 discusses errors that occur when users block cookies from being stored in their browsers.

### 2.3.1. Blocking <script>

Modern web browsers can install a program (called plugins) that will block <script> from being initiated while a webpage is being rendered. Blocking <script> is the web tracking equivalent to refusals in ordinary survey work, as client side on-site data collection relies on <script> to send a pixel request in order to collect data on visitor behavior. No data will be collected on visitors that block <script> and if the behavior of these visitors differ from the behavior of visitors that allow <script>, then the estimates produced on visitor data will be biased.

In most cases, installing and activating programs that block <script> is a conscious act, which implies that all users who choose to block <script> have something in common, and something that sets them apart from all other users that allow data to be collected. There are several potential reasons for installing and activating programs that block <script>. One is that the user does not want his or her online behavior to be tracked by website owners. Much content that can be interacted with on websites, like commercials, are often managed using <script>, and the intention of blocking these commercials (et cetera) might be another reason for a user to use <script>-blocking software.

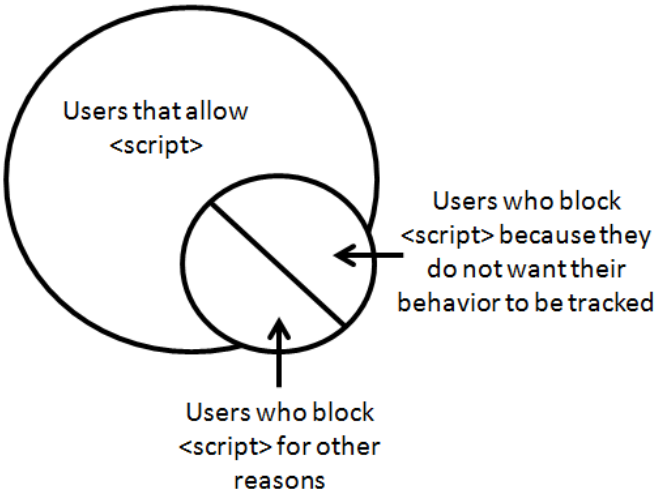


*Figure 14: Two types of users that block <script>*

It is not possible to distinguish between the two groups of users that choose to block <script>, as no data is collected on either. Neither is it possible to determine exactly how many of the total number of visitors it is that block <script>. Attempts have been made, however, estimating 1-2% of all visits to be from visitors who have disabled JavaScript (Zakas 2010). <script>-blocking is a very serious issue, since having the size of the group of nonrespondents set to being unknown severely limits the possibility of making inference from the data that is being collected on all the other visitors to the website.

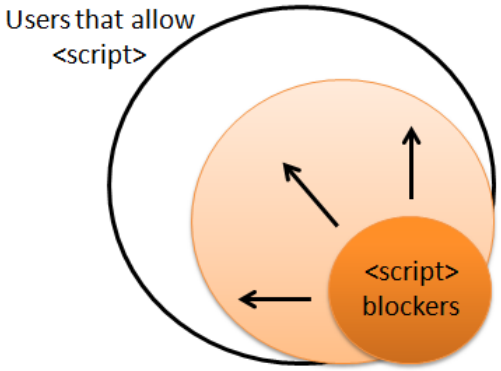


*Figure 15: The amount of <script> blockers is unknown*

A survey could be conducted to determine how many Internet users there are that use <script> blocking software. It would have to be a continuous survey in order to maintain validity. The survey would benefit from trying to establish nonresponse benchmarks for different types of websites and also for different countries and Internet user groups.

A usability study could be conducted; the results from the study could be used to model how <script> blockers behavior differs from those of ordinary users that allow <script>. It is important to be careful when interpreting the results from this study, however, since the controlled environment of a usability study could generate a bias regarding the testers' behavior.

By setting up server side on-site tracking (i.e. using server logs instead of JavaScript), a separate registry will be created that stores data on visitors' behaviors. It would then be possible to match this register with the database storing data for client side on-site tracking (<script> tracking) and approximately determine how many visitors there have been to the site that blocked <script>. Even though very simple server log tracking could be set up and still produce reliable data for this purpose, this solution to the error issue is costly due to the fact that the companies need to invest in a completely new tracking solution.

Another, possibly superior, solution to setting up server logs would be to incorporate <noscript> tags together with <script> tags in the source code for the web pages. <noscript> is initiated when <script> fails to initiate, and as such they are an automatic backup to failed <script> initiations. Setting up <noscript> could be done in two ways. Either the <noscript> tags are designed and implemented to complement their corresponding <script> tags (i.e., the <noscript> tags are coded to collect the same behavior data as the <script> tags) or the <noscript> tags are designed to collect rudimentary visitor behavior data.

Example of rudimentary behavior data in this case could be the page that the data was collected on, the previous page for the visitor, and whether the visitor is actively blocking <scripts>. As such, using the simple <noscript> solution would enable the site owners to determine the amount of visitors that come to the site and block <script>, but more advanced behavior analyses would not be possible. This simple <noscript> solution

would be easier to handle for company programmers than the advance <noscript> solution, where the <noscript>tags are designed to complement the <script> tags on every page. All data that can be collected by using <script> can also be collected by using <noscript>. The programming code required is however more tedious to work with for programmers, which is the most common reason why this solution is not broadly used. It requires much resources and time for programmers to manage these <noscript> tags because the <noscript> would be hardcoded. The common opinion is that for as long as <script> blocking is not a major issue then resources might as well be distributed elsewhere and programmers can prioritize other programming projects, rather than working with <noscript> backup coding.

The simple <noscript> solution helps companies and organizations determine whether <script> blocking is a major error source for their website. Weighting the cost for implementing the simple <noscript> solution to the information that can be gained from implementing it, it is recommended for most organizations and companies to implement this <noscript> solution.

### 2.3.2. Blocking cookies

It is possible to configure modern web browsers and install a program (commonly referred to as plugins) that will block cookies from being saved onto the browser. As cookies are used to identify returning visitors to a website, blocking cookies makes a visitor being counted as a new visitor every time he or she comes to the site, even though he or she might be a returning visitor. There is no way for the tracking solution to recognize the returning user if there is no cookie stored in the browser.

Users who block cookies will generate skewed estimates for site popularity and visitor return frequency for the site. For example, if a user comes to a website twice a day for a week but blocks cookies the entire time, instead of showing statistics of one visitor visiting the site 14 times over the course of 7 days, data will say that 14 different visitors have visited the site during those 7 days. There will be an overestimation on how many people have visited the site (14 instead of 1), and an underestimate of how many times visitors return during a week (0 instead of 13).

The severity of this error is mostly dependent on the behavior of the users that are blocking cookies. Firstly, websites that attract users that are prone to blocking cookies will also accumulate more biased estimates of visitor frequency. Secondly, if the cookie blocking user is a frequent visitor to a website, then estimates will be more biased than if the cookie blocking user is an infrequent visitor to a site. Figure 16 provides an illustration of this.

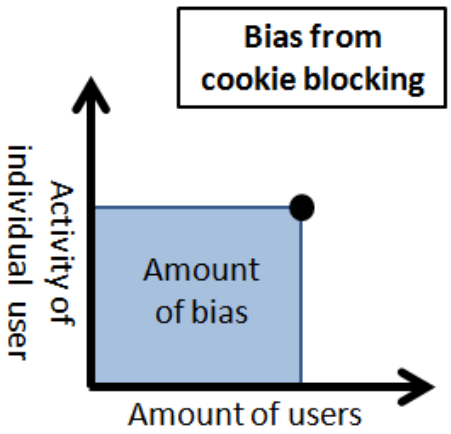


*Figure 16. Amount of bias as a product of the number of users blocking cookies and their individual activity.*

This error source can be addressed by conducting studies that would help gain an understanding regarding how many of the users that block cookies and studies that evaluate the cookie blocking users' behaviors and see if they behave differently than non-blocking users. These studies could range from surveys to usability testing to focus groups, depending on what particular aspect of this error the study is addressing.

A technical solution that counts the number of failed cookie placement attempts from the server could be developed. The activities of the cookie blocking users would still be unknown, but at least one part of the problem would be assessed.

## 2.4. Measurement error

Measurement errors occur when the respondent gives faulty information either intentionally or unintentionally. In ordinary survey work measurement error is usually a major error source. The main reason for measurement error in interview studies is because the respondents have to phrase their answers to the interviewer, and there is a risk for miscommunications due to for example language barriers or body language (Biemer 2010).

Measurement error takes a different form for online behavior data collection as the 'respondents' are being tracked instead of consciously providing information to the data collector. In other words, client side on-site data collection is latent and potentially unknown to the visitor that is being tracked. As such, the main source of measurement error has been moved from the respondent to the data collector and the data collection tools used for tracking the visitors to the websites. All actors are collectively responsible for data collection, but it is the programmer that designs and constructs the code that will collect the visitor behavior data.

Section 2.4.1 will discuss errors that occur due to erroneous data collection source code implementation. Section 2.4.2 will look at errors that occur when visitors' are using two web browser tabs simultaneously. Section 2.4.3 will go over errors that occur from users'

tendency to delete cookies stored in their browsers, and section 2.4.4 will discuss measurement errors that occur because people borrow each others' computers.

### 2.4.1. Tagging the site in the wrong way

Client side on-site tracking requires programmers to write the code and implement it on web pages for visitors' behavior data to be successfully collected. If the code is written incorrectly or if it is implemented incorrectly on the pages then faulty data will be collected or no data will be collected at all. The most usual source of this error is programmers misspelling when designing the data collection code. The reasons for misspelling vary, however, ranging from careless and sloppy programming to unclear instructions and requirement documents to coding errors due to programmer inexperience.

This error can be addressed by creating routines and checklists that help to quality control check code construction and code implementation. Examples of this could be to appoint assisting programmers to check their colleagues' data collection codes, or have test environments and testers that test that the codes are functioning properly after being implemented.

When data collection programming code errors or implementation errors occur, consequences are severe. Yet, it is not unheard of that a programming error slipped through the quality control check. It is important to educate company management and managers to allocate adequate resources to quality checking data collection programming code and implementation.

### 2.4.2. Inconsistent visitor movement pattern due to multiple tabs usage

Modern web browsers have the option to have multiple tabs open and active at the same time. A tab is a browser window within the browser window. Using tabs enables the user to be on two or more web pages at the same time. These pages can either be on the same website or on different sites.

If the visitor is using two tabs to visit two pages on the same website, then data collected on the visitors' previous pages will be inconsistent and faulty. For example, a visitor is visiting page A on website M in one tab and page X on website M in a second tab. From page A the visitor can click to go to page B, but not to get to page X or Y. From page X the visitor can click to get to page Y, but not to go to page A or B.

If the visitor comes to both page A and X by typing their URLs, first for page A in tab 1 and then for page B in tab 2, then the client side on-site tracking solution will register that the visitor went from page A to page X. If the user then goes back to tab 1 and clicks to go from page A to page B, the tracking solution will register that he or she went from page X to page B. Then, if the user goes back to tab 2 and clicks to go from page X to page Y, the tracking solution will register that he or she went from page B to page Y. An illustration of this example is presented in figure 17.

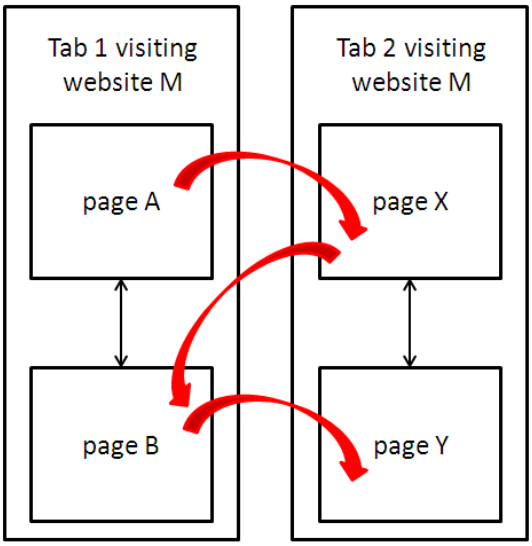


*Figure 17: Black arrows indicate how the visitor can navigate on the website. Red arrows indicate the described visitors' movement pattern.*

While this may be the users' actual movement pattern, it can become hard to analyze for the data analyst because it is illogical (since the site is only designed to enable the visitor to move from A to B, not from A to X) – potentially leading the data analyst to doubt the correctness of the data.

The error imposed by this error source is best redeemed by educating the data analysts. If the analyst is aware of the potential for this illogical visitor movement pattern, then he or she can take this into account while analyzing data and view the odd behavior as anomalies rather than taking it for faulty data. This is based on the assumption that very few people use multiple tabs to browse the same website. For websites where this attitude is more common, there will be an increased amount of suspect and faulty data that the analyst will have to address.

### 2.4.3. Deleting cookies

Cookies are stored in a visitors' browser when he or she comes to a website which is being tracked. This cookie is used for recognizing a returning visitor to the site. If a user manually deletes all his or her cookies then the user will be considered as a new visitor when returning to the website the next time. This will result in an overestimation of how many visitors have visited the site.

For example, users A and B come to a website every second day during a period of ten days (i.e., five times each during these ten days). On day 4 and day 8 user B deletes the cookies in his or her browser. Web tracking will register five visits from one visitor for user A and two plus two plus one visits from three different visitors for user B. An illustration of this example is presented in figure 18.


*Figure 18. User A and user B visit an equal amount of times, but user B is registered as three different visitors.*

One example reason for a user to delete his or her cookies could be that the he or she once saved a password to be automatically filled in by the cookie whenever the user comes to a site, but that the user has now changed his or her mind and does not want the password to be automatically filled in anymore. Deleting cookies and deleting the browser history (i.e., deleting the list of all websites and web pages that the browser has viewed) are often done simultaneously, so when a user deletes his or her browser history the cookies will be deleted as well.

Browsers can be set to delete cookies automatically, for example every fortnight or every time the browser window is closed. An example reason for setting cookies to be deleted automatically could be that the user wants passwords for websites to be saved during the session when he or she is using the device, but not once he or she is done with the session.

In order to understand why users delete their cookies, and how often, a survey could be conducted. The results of such a study would not help limiting the visitor overestimation caused by users deleting their cookies, but the knowledge gained from the study might help developing new methods for handling this error source.

### 2.4.4. Borrowing computers

In ordinary survey work, filling out questionnaires for others is a common measurement error that is very hard to control for the data collector. As web tracking is unable to determine what person is actually using a device, borrowing another persons' computer, tablet or SmartPhone and using it to browse the Internet is the web tracking equivalent to this measurement error. The web tracking will not be able to determine that it is another person than intended that is browsing the websites.

As with the case of screen sharing (see section 2.2.3), it is impossible to distinguish what person it is that is actually sitting in front of the monitor, and because of this it is impossible to erase this error. The bias imposed on estimates is limited to individual data inference, however, as overall site averages generally do not take individual users' behaviors into specific account, but instead look at all visitors to the site together. Site averages that are calculated totals or calculated on a visit basis do not suffer any bias from this error. Examples of non-suffering averages are *page depth per visit* and *average time spent on site*. Site averages that are calculated on a visitor basis will suffer bias, however.

An example of a site average that suffers from this error is the number of *unique visitors* to the site. Another example of web data that suffer from this error is campaign attribution (i.e. what promotional campaigns that can be linked to the visitors' browser), as campaigns can be erroneously attributed to the ordinary users' browser because the borrowing user activated the campaign, even though the ordinary user would not consciously activate the campaign himself.

A focus group could be put together or a survey could be conducted to ask people how often they borrow devices and their attitude towards lending and borrowing devices.

## 2.5. Data-processing error

Data-processing errors are all errors related to data-processing of any kind. Examples of this in ordinary survey work include coding of questionnaire responses (this is not to be confused with writing source code for websites), editing odd or missing values in the data repository or performing data transformations (Biemer 2010).

Out of the five types of nonsampling errors, processing error is the type of error that has the most ties between ordinary survey work and online behavior tracking. Many of the potential computational errors are the same, like summations and transformations of data that are done to prepare data for analysis. Every time an analyst works with data manually, the human factor imposes risk of errors being made.

In online behavior tracking data analysts are responsible for the majority of the processes related to data management and data manipulation, but programmers also play a crucial role in establishing the data collection framework and assuring that data will be processed correctly both while being collected and while being stored in databases.

Section 2.5.1 will discuss errors that can occur during the extraction process when data is extracted from a database. Section 2.5.2 will discuss the importance of distinguishing between data sources when collecting online behavior data.

### 2.5.1. Data extraction

It is common for client side on-site tracking that the database storing the collected data is managed by the company that is also providing the tracking solution. For example, Adobe store data that is collected through their data collection solution (Omniture 2010). For a website owner, it is often not possible to go into the database and edit or remove any data. Data analysts mostly work through reporting tools provided by the tracking solution provider, but it is also possible to make extractions from the databases (Dykes 2012). How data is extracted (in what file format, for example) depends on the tracking solution. For some solutions multiple extraction formats are available.

An example error that can occur during data extraction is that the extraction tool might not use the same type of punctuation as the data management tool that the analyst will use

to work with the data. For example if the extraction tool is using a comma to separate between thousands while the data management tool uses the comma before decimals, then the extraction process will erroneously transform 1024 into 1,024. Further, hundreds will not be affected (634 will remain 634 during extraction), but a thousand that ends with a zero will lose that zero in the transformation (1420 will erroneously transform into 1,42).

It is important that analysts have good experience with the data extraction tools and the data management tools that they are working with. It is also important to educate the analysts on how to spot errors when they occur and how to correct them. Further, the web tracking service providers could work to harmonize their extraction tools to work well with common data management tools like Microsoft Excel in order to offer a simple and robust service to the analysts working with their extraction tools.

### 2.5.2. Keep separate data for different website designs on different devices

Many companies and organizations design different versions of their websites for different devices, which leads to different experiences for the companies' customers depending on through which site they interact with the company.

If data for these different sites is collected to the same database without being marked from which site design and device type it was collected, then analysis will be performed on mixed data. Different site designs stimulate different behaviors from visitors, so it is very important to distinguish the sites data. If data is stored without being labeled, distinctive behaviors for one device might skew the analysis and decision-making for another device. For example, visitors that are visiting the tablet version of the site might be more prone to look at products than visitors that are using SmartPhones who use the mobile site for locating nearby stores for the company. Without correct labeling, analysis could prompt the company to invest resources into improving the product pages design for the mobile device, even though these users would much prefer resources being invested into the store locator tool.

It is important that all actors understand why this site-device labeling is important, and that resources are allocated so that this part of data processing is not ignored. Programmers need education on the importance of this labeling, while stakeholders and data analysts need education in how much work there is for a programmer to enable this labeling. Further, much can be gained from establishing routines for data quality control that enables workers to check that data that comes into the databases are labeled correctly. Both these actions are costly for the organization, however, as they divert resources from the companies' main business.

# 3. Conclusions and future work

This thesis strives to emphasize the importance of data quality of Big Data. It does so by evaluating data quality of client side on-site online behavior clickstream data.

By applying the Total Survey Error framework to client side on-site web tracking, a rather large number of potential error sources have been detected. The bias imposed by these error sources varies from large to small, but the major takeaway is the large number of error sources actually identified. It is not unreasonable to believe that this substantial amount of error sources is not unique for this type of data collection. Rather, it is a fact that many error sources can be identified in all types of data relevant in a Big Data context (for example financial data, social media data, astronomy data, city administration data, to name a few).

It is important to remember that the whole is made up of the pieces. The variety of interconnected data sources is one of the defining elements of Big Data. Each of these sources are, in turn, made up of collected data points, and if these data points suffer from some type of systematic error then this bias will not vanish when they move through the hierarchy. In the worst case scenario a big scale analysis can come to the wrong conclusions because so many of the pieces it bases its analysis on are of low quality. It goes without saying that this scenario can be costly for the company concerned. Thus it is very important not to neglect quality work.

Total Survey Error could be applied for other types of designed data, but also for organic data. Each data collection method would have its own concepts to handle. For example, specification error and measurement error can be more severe for designed data, while data processing error can be worse for organic data. Total Survey Error is still a versatile framework to use for this data quality evaluation. It would require quite an effort to apply Total Survey Error in this context but it would certainly be worthwhile.

This thesis serves only as an introduction to what can be done using statistics within the field of online behavior data collection. Throughout Section 2 the author presents and provides examples of each of the five nonsampling error types. What the author has done is, however, only scratching on the surface on what can be accomplished in this field.

Sampling error sources could be addressed. In traditional survey methodology sampling errors are errors that occur due to the sampling scheme, the sample size and the choice of estimators for the survey. As for web tracking methodology other concepts apply, and research could be conducted on determining these other concepts and evaluating them. Even though sampling is unusual in client side on-site tracking, much can be gained from establishing what types of sampling schemes are viable, and to determine what errors can occur while performing sampling of website visitors.

There are many error sources discussed in this thesis that could be addressed and evaluated by a survey. Examples of topics in a survey include:

- User attitudes regarding script blocking
- User attitudes regarding cookie blocking and user behavior regarding cookie deletion
- User behavior regarding multiple browser tabs
- How users behave differently while sharing screens with someone compared to when they are browsing on their own

How many devices the users own and use, what social group they belong to, and what types of websites they usually visit (and why) would also be interesting to examine, especially when the data can be correlated with the behavior data.

The results of such a survey would be valuable in itself from an academic point of view, but would also be very useful for companies and businesses that are active within the web analytics industry. In the best case scenario, the results from such a survey could be incorporated into the web tracking data collection tools, resulting in more accurate data collected by these tools.

Also, conducting other qualitative research on how people behave on the Internet could be beneficial. The results of this research could then be incorporated into the analysis of web tracking data. Even thought this would be done ad hoc it is still much better than not taking this knowledge into account at all. Better still would be if the results of this research could be incorporated in the most common reporting and data extraction tools that are available to analysts today.

This thesis investigates only nonsampling errors for client side on-site tracking, which is part of Total Survey Error. Further research could be conducted by applying the broader Total Survey Quality framework on client side on-site tracking. This would evaluate process quality and determine areas that could be improved.

One of the consistent themes of this thesis is that other online behavior data collection methods are able to fill in where client side on-site tracking is lacking. Very often web tracking can benefit from being combined with other methods. Analysts of online behavior data would benefit from actively working with different methods and tools that complement each other, since being restricted to a single method or tool is limiting. It is limiting in the sense that many error sources are hard or even impossible to address and correct for when using a single tool only.

The purpose of collecting data is to use it. Data that is collected but not used is a waste of resources because it steals resources from other projects. It is important to have a plan for designed data, but also for organic data, so that no data is left unused and forgotten. By having a larger plan the value of organic data will increase as well. After all, successful use of Big Data requires data of high quality, a purpose, as well as the means for working with and analyzing the data. If any of these ingredients are not up to par, then the whole process is not up to par either. This is why continuing evaluation of data quality is so important.

# *References*


Andersson, D. (2012). Hur många människor motsvarar X unika besökare? [online]. (How many people are X unique visitors?) *Available from: http://www.outfox.se/hur-manga-manniskor-motsvarar-x-unika-besokare/* [Accessed 20 May 2013]. (in Swedish)

Biemer, P. (2010). Total Survey Error: Design, Implementation, and Evaluation. *Public Opinion Quarterly, Vol. 74, No 5, pp. 817-848.*

Champkin, J. (2012a) Big data, big issues (Editorial article). *Significance Magazine, August 2012, Volume 9, Issue 4, pp. 2.*

Champkin, J. (2012b) From big data to the White House (Editorial article). *Significance Magazine, August 2012, Volume 9, Issue 4, pp. 2.*

Creswell, J. (2012). *Qualitative Inquiry & Research Design: Choosing Among Five Approaches,*. Third Edition. SAGE.

Couper, M.P. (2000). Review: Web surveys: A review of issues and approaches. *The Public Opinion Quarterly, 64(4), 464-494.*

Dykes, B. (2012). How to Get Data In and Out of SiteCatalyst - Part II [online]. *Available from: http://blogs.adobe.com/digitalmarketing/analytics/how-to-get-data-in-and-out-of-sitecatalyst-part-ii/ [Accessed 23 May 2013]*.

Eurostat. (2013). Table: Information Society Statistics > Computer and the Internet in Households and Enterprises > Internet - Level of Access, Use and Activities > Level of Internet Access - Households [online]. *Available from: http://epp.eurostat.ec.europa.eu/portal/page/portal/information_society/data/main_tables [Accessed 7 May 2013]*.

Feigelson, E., and Babu, J. (2012). Big Data in astronomy. *Significance Magazine, August 2012, Volume 9, Issue 4, pp. 22-25.*

Flanagan, D. (2011). *JavaScript: The Definitive Guide*, Sixth Edition. O'Reilly Media.

Google. (2013). How to set up the web tracking code [online]. *Available from: https://support.google.com/analytics/answer/1008080?hl=en [Accessed 23 May 2013]*.

Gill, M., and Wigder, Z.D. (2013). *European Online Retail Forecast, 2012 To 2017: Economic Instability Across Europe Will Do Little To Slow Retail eCommerce Growth.* Forrester Research.

Goldsmith, R.F. (2004). *Discovering Real Business Requirements for Software Project Success.* Artech House.

Gourley, D., Totty, B., Sayer, M., Aggarwal, A., and Reddy, S. (2002). *HTTP: The Definitive Guide.* O'Reilly Media.

Groves, R. (2011). "Designed data" and "organic data" [online]. *Available from: http://directorsblog.blogs.census.gov/2011/05/31/designed-data-and-organic-data/ [Accessed 15 May 2013]*.

Halevi, G., and Moed, H. (2012). The Evolution of Big Data as a Research and Scientific Topic: Overview of the Literature. *Research Trends, Issue 30, September 2012, pp. 3-6.*

Hargis, G., Carey, M., Hernandez, A. K., Hughes, P., Longo, D., Rouiller, S., and Wilde, E. (2004). *Developing quality technical information: A handbook for writers and editors.* IBM Press.

Hilberg, M. (2012). How much information is there in the "information society"? *Significance Magazine, August 2012, Volume 9, Issue 4, pp. 8-12.*

IBM. (2013). Big data spans four dimensions: Volume, Velocity, Variety, and Veracity [online]. *Available from: http://bigdatafoundation.com/blog/big-data-spans-four-dimensions-volume-velocity-variety-and-veracity/ [Accessed 22 May 2013]*.

Internetstatistik.se. (2012). Nästan 1,2 miljarder aktiva mobila bredbandsabonnemang globalt [online]. (Almost 1,2 billion active mobile broadbands globally.) *Available from: http://www.Internetstatistik.se/artiklar/12-miljarder-aktiva-mobila-bredbandsabonnemang/ [Accessed 7 May 2013]*. (in Swedish). Secondary source from ITU (International Telecommunications Union).

Internetstatistik.se. (2013). Kvarts miljard registrerade domännamn [online]. (Quarter of a billion registered domain names.) *Available from: http://www.Internetstatistik.se/artiklar/kvarts-miljard-registrerade-domannamn/ [Accessed 7 May 2013]*. (in Swedish.) Secondary source from Verisign.

Juran, J.M., and Gryna, F. (Eds) (1988). *Juran's Quality Control Hanbook*, Fourth Edition. McGraw-Hill.

Kaushik, A. (2009). *Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity.* Sybex.

Keller, S.A., Koonin, S., and Shipp, S. (2012). Big Data and city living - what can it do for us? *Significance Magazine, August 2012, Volume 9, Issue 4, pp. 4-7.*

Laney, D. (2001). 3-D Data Management: Controlling Data Volume, Velocity and Variety. *META Group Research Note, February, 6.*

Lansdall-Welfare, T., Lampos, V., and Cristianini, N. (2012). Nowcasting the mood of the nation. *Significance Magazine, August 2012, Volume 9, Issue 4, pp. 26-28.*

Lorentz, A. (2013). Big Data, Fast Data, Smart Data [online]. *Available from: http://www.wired.com/insights/2013/04/big-data-fast-data-smart-data/ [Accessed 23 May 2013]*.

Lyberg, L. (2012). Survey Quality. *Survey Methodology, December 2012, Vol. 38, No. 2, pp. 107-130.*

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Hung Byers, A. (2011). *Big Data: The next frontier for innovation, competition, and productivity.* McKinsey & Company.

Mastrangelo, T. (2012). The Big Deal about Big Data [online]. *Available from: http://blog.advaoptical.com/the-big-deal-about-big-data/ [Accessed 22 May 2013]*.

Mayer-Schonberger, V., and Cukier, K.N. (2013). *Big Data: A Revolution That Will Transform How We Live, Work, and Think.* Eamon Dolan/Houghton Mifflin Harcourt.

Micklethwait, J. (2010) The Data Deluge (Editorial article). *The Economist February 25th – March 5th issue 2010. Also available online from: http://www.economist.com/node/15579717 [Accessed 15 May 2013]*.

Mulpuru, S. (2011). *US Online Retail Forecast, 2010 To 2015: eCommerce Growth Accelerates Following "The Great Recession".* Forrester Research.

Musciano, C., and Kennedy, B. (2006) *HTML & XHTML: The Definitive Guide,* Sixth Edition. O'Reilly Media.

Omniture. (2010). SiteCatalyst Implementation Manual [PDF online]. *Available from: http://www.markcregan.com/wp-content/uploads/2010/06/sitecatalyst_implementation_guide.pdf [Accessed 23 May 2013]*.

Palmer, C. (2013). Amex to tap Big Data to expose fake reviews [online]. *Available from: http://www.itnews.com.au/News/342993,amex-to-tap-big-data-to-expose-fake-reviews.aspx [Accessed 15 May 2013]*.

Peterson, E. (2006). *The Big Book of Key Performance Indicators.* Web Analytics Demystified.

Rasmussen, M.J. (2012) European Union ePrivacy Directive – Latest Update [online]. *Available from: http://blogs.adobe.com/digitalmarketing/executive-insights/european-union-eprivacy-directive-update/ [Accessed 20 May 2013]*.

Rencher, B. (2013). The Last Millisecond (keynote presentation) [online video]. Adobe EMEA Summit 2013. *Available online from: http://summit-emea.adobe.com/online2013.html#tv [Accessed 15 May 2013]*.

Rubin, J., and Chisnell, D. (2008). *Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests*, Second Edition. Wiley Publishing.

Ryen, A. (2004). *Kvalitativ Intervju, från vetenskapsteori till fältstudier.* (Qualitative interviews, from theory to practice.) Liber. (in Swedish.)

Schaefer, K., Cochran, J., Forsyth, S., Glendenning, D., and Perkins, B. (2012). *Professional Microsoft IIS 8,* First Edition. Wrox.

Schmitt, B.H. (2003). *Customer experience management: a revolutionary approach to connecting with your customers.* Wiley.

Smith, M.L., and Erwin, J. (2007). Role & responsibility charting (RACI) [PDF online]. *Available from: http://myclass.peelschools.org/sec/12/4268/Resources/RACI_R_Web3_1.pdf [Accessed 20 May 2013]*.

Spencer, D. (2009). *Card Sorting: Designing Usable Categories.* Rosenfeld Media.

Zakas, N.C. (2010). How many users have JavaScript disabled? [online]. *Available from: http://developer.yahoo.com/blogs/ydn/many-users-javascript-disabled-14121.html [Accessed 20 May 2013]*.

ZOHO Corporation. (2012). EventLog Analyzer 8 User Guide [PDF online]. *Available from: http://www.manageengine.com/products/eventlog/eventloganalyzer-userguide.pdf [Accessed 23 May 2013]*.