

# Bugs in your shopping cart: A Taxonomy

Giri Vijayaraghavan  
Cem Kaner  
Florida Institute of Technology

**Presented at the International Software Quality Week, San Francisco, CA, September 2002.**

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

## Introduction

Imagine being asked to test a web site's shopping cart. If you hadn't tested one before, where would you start? What experience would you draw on? Where would you look for more information? Even very experienced testers have blind spots when they try to generate test ideas for an application that they have not tested. This paper presents a simple outline that will help you generate test ideas and limit your blind spots. The outline is the result of a year's research on classifying e-commerce-related failures and risks. The result has 60 top-level categories and examples of errors (potential issues to test for) under most categories. In many cases, we also link to examples of e-commerce defects that have been publicized in the press.

## Goals of this outline

Using the list, you could pick a category of interest (such as accessibility or software upgrade), read descriptions of several types of problems that fit within that category, and so identify a few issues that would be appropriate to test for in your application. Based on feedback to the authors of *Testing Computer Software*, we believe that many testers will be able to use this list to identify potential problems that they would otherwise have missed.

We intend the outline to serve similar functions to Kaner / Falk / Nguyen's bug appendix in *Testing Computer Software* (TCS):

- Help testers generate ideas;
- Help test plan inspectors check a set of tests for thoroughness and coverage;
- Help testers and other stakeholders identify risks during discussions of prioritizing the testing effort.

Kaner expects this to become the successor to the TCS list, and we have structured the outline (e.g. more top-level categories and fewer levels) based on feedback from many TCS readers

## The nature of e-commerce testing

"E-commerce testing involves testing high value, high risk, and high performance business critical systems"[1]. The cost of a failure in an e-commerce system is hence formidable and generally the testing process involves considerable effort in designing and integrating an effective risk management process into the testing to avoid or minimize the cost of failures.

The distributed nature of the underlying e-commerce applications also adds to the complexity in developing a comprehensive risk based testing approach because it makes it difficult to exactly pinpoint where the risk of failure exists. Which application failed? Where did it fail? And when did it fail? E-commerce testing only involves testing the individual underlying components and web applications for potential risks and failures, along with the e-commerce “web site” as a whole.

A bug in an underlying vendor specific component such as a Web server, transaction server or database system may be missed or ignored because the focus of testing might be on the site as a whole or on the code specifically written for the web application. Unfortunately, an inconspicuous bug in a third-party component might cause a serious failure when the system is in production, leading to bad publicity, and lost reputation, customer confidence, and revenue.

## **The problem of transitioning to e-commerce testing from other platforms**

E-commerce testing involves numerous challenges, such as new or unfamiliar technologies, or technology-driven business processes or logic that the tester doesn't understand how to test. Learning the necessary information or skills may be complex but can be simplified with strategies for quickly generating test ideas or test plans that address potential risks. We believe that an outline of well-researched potential failures can help flatten the steep learning curve involved with e-commerce testing.

It may be worthwhile for any tester who is moving into the e-commerce-testing arena to remember the following issues:

- **“Test Global and Test Distributed”**: E-commerce systems are truly global in spirit and structure. The different underlying systems may be on different continents, but they appear to integrate seamlessly over large, distributed and non-homogenous business networks and other communication channels. Risk analysis and test planning should allow for potential problems caused, for example, by a side effect of a software upgrade on a transaction server physically located in Holland on a transaction taking place between a customer in Korea and a retailer in the US.
- **3 Ms where 1<sup>st</sup> M: Multiple Platform 2<sup>nd</sup> M: Multiple Clients (Browsers) 3<sup>rd</sup> M: Multiple customer profiles**: This will be a considerable change for a tester who comes to e-commerce testing from other traditional testing fields. In the stand-alone or older client servers, the user's platform, the client type and the nature of the user might be well known to the developers and testers. E-commerce systems involve more uncharted territory, with greater diversity of operating systems, browsers and other system software and hardware. The user profile varies greatly in terms of age, gender, taste and usage-testing business software has become as configurationally complex as testing

consumer software, but with more serious consequences in the event of failure.

- ❑ ***Learning to count in “web years”***: Changes and updates are inherent in E-commerce sites. Content and target platforms change quickly, without much time for planning and regression testing of each change. This can be challenging for testers coming from traditional business applications. The e-commerce tester must learn to generate effective sets of test ideas rapidly.
- ❑ ***The risk of testing in a “not-so-representative test environment”***  
In an e-commerce world, creating a completely representative test environment is often impossible. With a limit on how much can be actually simulated in a test lab, there is real risk of not knowing how the application will behave in some environments in the field.

## How to use this Outline?

- ❑ ***Generate new test ideas:***  
The outline provides about 60 top-level categories with examples of errors under each one of them. The categorized outline inspires test idea generation if the tester considers the function-under-test, and then considers how the function would fail with respect to one of the categories.
- ❑ ***Use the test ideas here, for applicable projects***  
This paper is a subset of a broader project that will be published in Giri Vijayaraghavan’s Master’s thesis. You will probably be able to find that thesis posted on the Net at [www.testingeducation.org](http://www.testingeducation.org) and [www.girivijay.com](http://www.girivijay.com) in January or February 2003. The thesis fills in details for more categories of shopping cart problems and less extensively considers failure modes for some other functions. We use the shopping cart as the example function because it has rich functionality (and thus many different failure modes). We use an example function to make the application of the categories more concrete. The same types of problems will show up in many other types of e-commerce application functions. Vijayaraghavan’s taxonomy is not exhaustive. We think the 60 top-level categories are sufficient, but below that level, you’ll be able to add plenty of your own examples if you use the outline. We recommend that you download Vijayaraghavan’s thesis and customize it as you use it on different projects. The more tailored it is to your company’s applications, the more long-term value it will have for you. If you do this, we ask that you send us any non-confidential additions that you’ve made, so that we can add them to a master list.
- ❑ ***Audit test plans by inspecting for tests for potential errors***  
It’s difficult to find the blind spots in a long test plan. There is so much detail available that it’s hard to see what’s missing. A list of potential failure modes provides you with an independent cross-check. To audit

the plan, consider each category in turn. Ask yourself whether failures within that category are *possible* in the software under test, and if so, what they might look like. Pick two or three possible failures and then check the test plan. Does it have tests that would catch these problems? If not, you've found a hole in the plan.

- ***As teaching material to assist in the training of testers who are new to e-commerce testing.***

The clear structure of the outline, the detailed list of possible risks and failures, and the concrete examples can help new testers broaden their understanding of the range of problems open to their discovery. You can also use them as anchoring points for discussions. For example, some test groups set up a weekly lunch meeting to improve their knowledge or skills. One of these lunches, you might talk about how to recognize accessibility errors. Another day, you might talk about how to recognize update-installation side effects. This outline can help you focus those discussions.

- ***As a presentation tool for explaining to managers the different types of failures that can occur in an e-commerce site.***

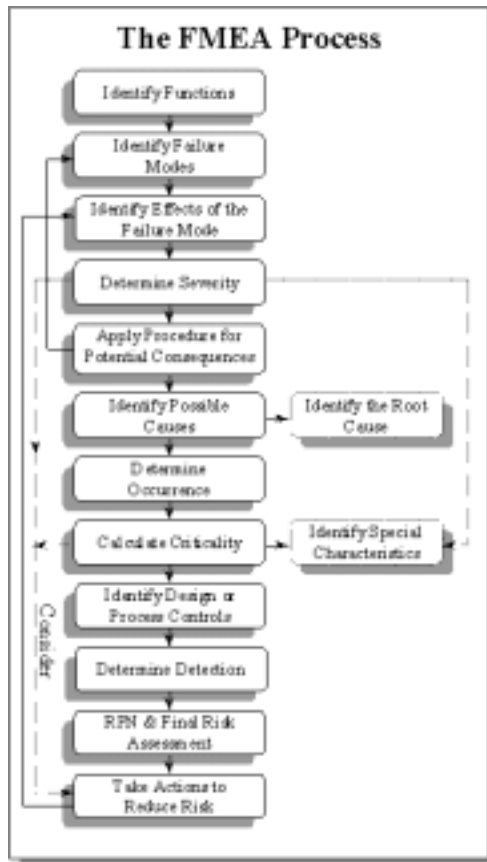
Managers don't necessarily understand the breadth of the scope of risks involved in e-commerce projects. Use this outline to develop a list of the types of issues that can be tested for. Cross-reference to published examples of errors that have shown up in the trade press (use ours or find your own). A well-organized presentation based on demonstrable risks can go a long way toward getting you adequate funding.

## **Failure mode and effects analysis (FMEA)**

This outline fits within the tradition of failure mode and effects analysis (FMEA). "FMEA has been widely used in the automobile, medical equipment, and aviation industries to focus development teams on potential failures and help them decide which risk are the most important to mitigate and what actions might be taken to mitigate them"[2].

Typically a FMEA process starts with identifying functions and identifying failure modes associated with the functions.

A typical FMEA process looks like fig 1.0:



Source: <http://www.fmeca.com/ffmethod/fmeaproc.htm>

We follow a similar approach in software: risk-based software test management involves identifying core system functions, identifying the different risks or failure modes associated with the functions, determining their severity, sometimes calculating their criticality and then prioritizing the risks to be addressed in the testing approach (Stale Amland, Risk-Based Testing and Metrics, EuroSTAR 99, Barcelona, Spain, 1999 [3]) FMEA differs from the risk-based test management approach in an important way--the FMEA approach often involves more careful attention to the fine details of the potential failure. In this respect, FMEA is more like the risk-based test design tradition, in which we work out the design details of tests based on the failure we are trying to detect (Cem Kaner, Black Box Software Testing course notes, available soon at [www.testingeducation.org](http://www.testingeducation.org).)

If FMEA were applied to an e-commerce site, it would involve:

***Identifying the functions***

Identifying functions and then generating the different function modes and failure modes is a tedious and time-consuming process, especially when we try to analyze a complex system such as a shopping cart driven e-commerce site. Decomposing an e-commerce site into constituent functions is further complicated by inter-dependencies between components and the overlapping architectures of e-commerce sites. My Master's thesis (Vijayaraghavan, in preparation) analyzes *User-Authentication (Login/Logout), Search, and Shopping*

*Cart* as a few of the functions in a shopping-cart-based, business-to-consumer e-commerce site.

### ***Identifying failure modes***

One of our main goals for the outline (and the thesis) is for it to serve as a brainstorming tool to help you generate the different failure modes associated with functions under test.

It's difficult to generate a comprehensive list of failure modes for a function for a variety of reasons:

- ❑ Lack of knowledge about all related and existing software bugs that have caused failures in the past or possess the potential to cause a failure in the future.
- ❑ Lack of knowledge about historical data on past failures and the causes behind them.
- ❑ Lack of an organized "Bug Taxonomy" that can remind the tester of patterns of potential failure/risk.

The sample list of "shopping cart" failure modes, which was generated using the outline provides references to known software bugs in e-commerce systems, provides historical data of past failures by citing issues publicized in the press.

## **The sample 'function': An E-commerce Shopping cart**

### ***An Introduction***

The familiar metaphor of a shopping cart that is present in many e-commerce sites has an important function of keeping track of the user's state while he/she is "shopping". A simple cart may just maintain a list of items that the user places in it and maintains state until the user finishes shopping and exits the system (by closing the browser). Complex and advanced carts have more sophisticated functionality such as real-time credit card processing and real-time order tracking.

As e-commerce sites grow in size and popularity, they tend to add more and more features to their shopping carts. Shopping carts have grown from simple state tracking functions to a highly sophisticated and creative piece of software offering a flexible range of user options.

The diversity, the creative imagination and technological innovation that have gone into the design of these next-generation-shopping carts, make them fascinating and challenging to test. Some issues that you may want to consider before testing a shopping cart are:

- ❑ The same issue of testing under "web years" that we discussed about general e-commerce testing applies here. Hence testers will have to reckon with testing under reduced time and sudden spikes in workload.
- ❑ A shopping cart may not scale. It may work well for 100 users but not for 1000. Testers should estimate performance standards early in testing.
- ❑ They are prolific and vary greatly in terms of design, size, complexity, and underlying technology. Hence no standard best practices exist that can

provide a single-point reference on how to efficiently and comprehensively test a shopping cart.

- ❑ Because of the rapid evolution and change in their design and features, it is common for testers to encounter legacy-shopping carts built with outdated technology.
- ❑ In contrast, the rapid changes in design and functionality might be a challenge to the less tech-savvy tester, as they add more learning pressure on the tester.

We thank Ms. Karen Johnson for sharing her experiences on testing a shopping cart and for her inputs on some of the issues/bugs mentioned in this paper.

## **Different types of shopping carts**

Testing each of the above shopping carts can be very different because of the difference in the way they have been built and hosted. Except for some **in-house shopping carts**, shopping carts tend to have a large number of 3<sup>rd</sup> party components, which are sometimes beyond the scope of the testing group.

**3<sup>rd</sup> party built and hosted shopping carts** are generally a cheaper alternative. They are employed by small-scale e-commerce sites with fewer staff to design and maintain the system. Sometimes the 3<sup>rd</sup> party host may be the same as your site-hosting provider. But 3<sup>rd</sup> party hosting wrenches much of the control away from the tester, since the bulk of the components are not in the tester's domain. But the positive side of this type of shopping cart is that the cart is smaller, less complex and generally easier to test.

**Out-of-the-box shopping carts** are customizable, pre-fabricated, and ready for deployment. Installation, customization, and configuration bugs are some of the common types of issues that testers will encounter when testing shopping carts of this genre. Though many serious security holes have been caught due to bad settings and bad configurations in out-of-the-box carts, much of the deeper level of testing becomes the responsibility of the vendor who originally developed the cart software.

And finally there are many **free shopping cart scripts** available for download in CGI or ASP script sites. Very simple and small-scale e-commerce sites tend to use these carts. From the tester's perspective, the risk is in the script code. While some of these scripts are well done, others are hastily developed by amateurs.

## **The Outline**

### **The Structure of the outline:**

This introductory material explains the structure of the outline, which is the core of this paper. The outline presented in this paper is a top-level list of about 60 risk categories and the *list of shopping cart failures* is a long and detailed list of different failures that can occur in a shopping cart under each category of the outline. The shopping cart failure list is an example of how the outline can be used to generate test ideas.

*Both the Outline and the list of shopping cart failures are work in progress. A more complete list will be available in the Master's thesis and later publications.*

The list of shopping cart failures has been categorized under the 60+ categories shown by the outline. Some categories have relevant sub-categories for ease of understanding and structure. We provide definitions and cite relevant literature for each category. The categories and sub-categories hold detailed lists of risks, errors, bugs, and failures that a tester might find when testing a shopping cart. Most of the categories also have examples of published bugs to illustrate the risks. Some categories also have thoughts on methods to test for this risk. More details on the qualitative categories can found in ISO 9126 [4]

### **Thoughts on how to use this list to generate test ideas for non-shopping cart applications**

We developed the list of shopping cart failures to study the use of the outline as a test idea generator. We think the list is a sufficiently broad and well-researched collection that it can be used as a starting point for testing other applications.

For example, the list contains a generalized collection of risks due to database failures (Database media failure, database statement failure, database instance failure and database-user process failure). This collection of database failures is relevant to all projects that use databases. Failures in web server, database server or cache server are not specific to just to a shopping cart driven e-commerce sites but to all web sites that use them. A tester testing a travel/airline ticket web site or an auction web site may find the list as relevant to his testing as a tester working on a shopping cart web site.

### **Outline: Table Format**

Performance
Reliability
Software Upgrade
UI Risks/Errors/Usability
Maintainability
Conformance
Adaptability
Stability
Analyzability
Operability
Understandability
Fault tolerance
Inter-operability
Accuracy
Internationalizability
Visibility
Accessibility
Localizability
Suitability
Compliance
Maturity



Recoverability
Learnability
Efficiency
Changeability
Testability
Installability
Replaceability
Insufficient Capacity Planning
Backend Connectivity failure
Bandwidth Sufficiency risks (During Peak)
Human Error
Calculation/Computation
Navigation Flow
Process Flow
Data flow
Transaction flow
User-System Interaction
Third-party Software failure
Database-Statement Failure
Database-Instance Failure
Database-User-Process Failure
Database-Media Failure
Memory failures/Memory Leaks
I/O Data Type Conflicts/Data Mismatch
Network risks
Functionality
Error Message /Exception Handling
Server-Side-Hardware Failure
Client-side-hardware failure
Third-party-hardware failure
Browser problems
Document Confidentiality
System Security
Client Privacy
Database Server failure
Cache Server Failure
Web-server failure
Transaction Server failure
ISP problems

## ***List of shopping cart failures***

### **Database Server failure**

A database server is software that manages data in a database. It updates, deletes, adds changes, and protects data [Network Magazine]. Database servers provide both the access

control and concurrency control. So while testing a shopping cart, if you find empty catalogs, unpopulated data fields and authentication problems, then you should check the database server. Some of the issues discussed here are based upon the discussion in the paper “Managing Database Server Performance within an Electronic Commerce Framework” [5] Here are different ways a shopping cart can fail, when the database server goes wrong:

- ❑ Inability to load or populate data in the product catalog.
- ❑ Inability to load or populate order data in the shopping cart.
- ❑ Inability to load or populate customer profiles.
- ❑ DB server failure may lead to a complete failure of data retrieval in the system since DB server manages/serves the data in the system.
- ❑ Increase in response time during "browse" transaction. Browse transaction generates high frequency, random, sequence of queries on the database server.
- ❑ The "shopping cart" transaction fails to update/load the billing details/price in the basket. Shopping cart transaction places medium weight, high frequency read/write operation.
- ❑ Increase in response time to load/update billing details, price lists and total in the basket.
- ❑ Failure or delay to commit the customer order to the database in the "Buy transaction".
- ❑ User-registration failure, unable to execute read-write process during user registration.
- ❑ Search process fails to execute since DB server failure may cause failure of read-only search process to fail.
- ❑ Increase in "search" time may indicate performance problems in Database server.

## **Cache Server failure**

Cache servers are used as intermediaries for web requests and retains previously requested copies of resources. The use of a cache server is to handle common requests locally and improve site performance by better speed up and reduced overhead on the web servers.

The common issues discussed here are can be found in more detail in “Known HTTP Proxy/Caching Problems [7]”

- ❑ Cache may return an outdated shopping cart document if the header is misrepresented or last modified date is omitted
- ❑ If shopping cart content is dynamic in nature, then cache server will not be able to serve new content
- ❑ If caching proxy server fails during shopping session, sometimes the browser fails to bypass server and may need to be reconfigured and shopping cart state may be lost
- ❑ If shopping cart uses any form of encoded response, the proxy might cache it and send it to a non-encoding capable client
- ❑ Sensitive shopping cart content may get cached by Interception proxies that break client cache directives like "No cache" or "Must revalidate"
- ❑ The cache server may end up blocking some methods used by the shopping cart software, because the method contained in the request is unknown to the proxy so instead it generates the default HTTP 501 Error as a response.
- ❑ Shopping carts that use IP address to track state of the cart, may fail because Interception proxies at ISP level may alter client's IP to that of the proxy itself
- ❑ A caching proxy mesh might break HTTP content serialization resulting in the user getting older content when the shopping cart page loads

### **Examples of related bugs and other known issues**

Is Web caching bad for the Internet?

<http://www.cnn.com/2000/TECH/computing/04/18/web.cache.idg/>

Known HTTP Proxy/Caching Problems

<http://www.wrec.org/Drafts/draft-ietf-wrec-known-prob-03.txt>

## **DATABASES**

The approach used here to classify the different databases can be found in more detail in "Oracle9i Database Administration: Recover Databases [6]", According to the definitions provided in it,

**Database statement-failure:** "Statement failure occurs when there is a logical failure in the handling of a statement".

**Database-Instance Failure:**

"Instance failure occurs when a problem prevents a database instance from continuing to run. An instance failure can result from a hardware problem, such as a power outage, or a software problem, such as an operating system crash. Instance failure also results when you issue a SHUTDOWN ABORT or STARTUP FORCE statement"

**Database-User-Process Failure:**

"A process failure is a failure in a user, server, or background process of a database instance such as an abnormal disconnect or process termination"

**Database-Media Failure:**

"An error can occur when trying to write or read an file on disk that is required to operate a database. This occurrence is called media failure because there is a physical problem reading or writing to files on the storage medium".

## Database statement-failure

- ❑ User may be attempting to issue a statement referencing a table in the shopping cart that does not exist
- ❑ A user may be attempting to issue a statement referencing a table in the product catalog, user database for which they do not have permission to access
- ❑ Flawed statement or Flawed query used by the web developer may make shopping cart data inaccessible to the user.
- ❑ Inability of a user to submit information, which is to be stored into a database because of inadequate table space allocation for the user/operation
- ❑ Flawed statement/query may lead to in-correct addition/deletion of items in the basket
- ❑ Incorrect access of tables may lead to incorrect computations/calculations of shipping/taxes
- ❑ Failure to clearly specify required fields, optional fields and edit permissions may lead to problems when data is being written back into the tables.
- ❑ Inefficient queries on the shopping cart tables

## Database-Instance Failure

- ❑ The number of simultaneous connections allowed is less than the maximum number required by the system for shopping cart transactions
- ❑ Power outage when shopping cart database is being accessed and no recovery routines exist.
- ❑ Check for issues where using a product database and multiple item forms together would cause an error

## Examples of related bugs and other known issues

### Database Glitches at Walmart.com

[http://www.internetnews.com/ec-news/article.php/4\\_739221](http://www.internetnews.com/ec-news/article.php/4_739221)

## Database-User-Process Failure

- ❑ Risk of user being unable to return to shopping cart after navigating away from the page since contents of cart not been saved
- ❑ User unable to add/delete/modify contents of the basket
- ❑ Client PC hangs during shopping cart transaction and user state not saved/retrievable
- ❑ Failure of the shopping cart database to rollback process on detection of user process failure

## Database-Media Failure

- ❑ Not enough memory on the system on which the shopping cart database resides
- ❑ Disk failures/Hard drive crashes, and other irreversible media corruption of the shopping cart database may cause complete loss of data
- ❑ Corruption of shopping cart Database backup

## Error Messages/ Exception Handling

Provided below is a detailed list of errors that you might encounter in an e-commerce site with a shopping cart and it might be useful to test for appropriate error messages. Testers should find this list useful to test a shopping cart site for error handling and check if the error handler handles these common errors. It has also been sub-categorized for ease of use on the basis of the kind of errors the system has been designed to handle.

- **Quantity error-handling**
  - ❑ Ability to erratically checkout an empty shopping cart and check if error message is displayed.
  - ❑ Ability to add negative numbers to the quantity field. Check for appropriate error handling
  - ❑ Accepts decimal entries for quantity but ignores the decimal point and either accepts the first or last digit alone, so 7.0 may be interpreted as 7 or 0 and no error handling exists to prompt or correct the error.
  - ❑ Accepts decimal entries for quantity but again ignores decimal point and accepts the quantity comprising of both the digits, so 7.0 may be interpreted as 70! And no error message to prompt or correct the error.
  - ❑ Quantity field not size-constrained and no error message to prompt user of acceptable values or data range.
  - ❑ An over-sensitive error handler may not let a user increase/decrease/edit the quantity field at an editable stage and may risk rendering the data entry final!
- **Information error-handling**
  - ❑ Forms requiring registration information, shipping address information, billing address information employ script based entry validation to validate entries but sometimes the scope of the script exceeds its limit and pops an error message for entry fields outside the limit of the script or optional fields.
  - ❑ Some address fields contain two parts, address 1 and address 2 in order to accommodate lengthy addresses. But some error handlers count both the fields as compulsory and pop error messages to users who leave address 2 empty (because their address is short and fits right into the first one!)
  - ❑ Long addresses may get clipped and no error message or routine exists to warn the user about the size constraint.
  - ❑ Lack of error routine to check for valid US zip code in the address section.
  - ❑ Check for trigger-happy error messages that sometimes pop up to a non-US shopper's dismay, to validate an empty US zip code.
  - ❑ Error message pops up informing the user of incomplete information entry but does not highlight the field where error exists.
- **Interaction and Transaction error-handling**
  - ❑ “*An Internal Server error*” may be displayed without any fix to the user, sometimes this error, which may be due to a missing term in the URL, can be fixed by appending a term, like say &reference to the address.
  - ❑ “*Inventory module error message*” may be displayed with no explanation to the user; sometimes this error occurs when two users access the last

item and the inventory control tries to update the order so that only one user gets access to the item.

- ❑ If you encounter an “*ODBC error* message“ when you click Checkout, you may be missing your "session ID", error handler should be enabled to handle this common error or should provide help to customers with simple fix to this errors.
- ❑ “*Timeout Error messages*” If any routine exists to check the time of inactivity and auto times out any shopping cart, such existence of timeout routines should be communicated to the user beforehand.
- **Payment/ Credit-Card error-handling**
  - ❑ An incorrect expiration date (be sure to use a two-digit year, such as "02") and supporting error message to prompt the user.
  - ❑ “*Invalid Card Number error message*”, if the card processing is a real time event in the cart, then user may be prompted to enter the number again or try a different card
  - ❑ Inconsistency between the address in the billing section and the address in the card. Check for user-understandable error message.
  - ❑ *Browser version too old to support card processing/secure protocols*, user must be pointed to the browser issue and not leave them in a limbo, with a clueless message pointing to card error instead of browser incompatibility issue.
  - ❑ *Invalid ABA Number error message*: If shopper is paying by check, he/she must supply valid checking account and "ABA" numbers.
  - ❑ *Site does not support the card used by the customer*; provide a message forehand about the type of cards the site supports.
  - ❑ “*Temporary Network Error messages*”: A temporary network problem may cause a data transmission error between the credit card processor and your bank.
  - ❑ Check if alternative error handling exists, when third party billing agents fail.
- **General error messages**
  - ❑ Unable to understand error message; Cryptic & undecipherable error messages especially in secure areas of the shopping cart may make users abandon their cart in panic.
  - ❑ A common mistake in a shopping cart error- handling system is displaying machine errors or compilation errors to the user instead of understandable error messages consistent with the language of the site.
  - ❑ Persuasive VBScript or Javascript error message boxes that pop up on an erratic entry but don't close on clicking OK!
  - ❑ Error-handling routine re-directs you to another page for explaining the error but provides no way to return back to the original state of the shopping cart.
  - ❑ Loads a pop-up error message box, but a 404 “page not found” error displayed in the error pop-up!

- ❑ *Over enthusiastic exception handling*: pops error message even after the error has been corrected or error message pops up for correct entries too due to failed script based validation routine.
- ❑ Error message box or an action to close the error box *causes illegal operation or illegal memory reference* in the browser software and causes the browser to close in the middle of a transaction.
- ❑ Error boxes written in scripts not supported or *incompatible with browser type*.
- ❑ *Typos, grammatical errors* in error messages that change the meaning of the intended error message
- ❑ *Illegible error message*: A combination of the color scheme of the message box and the font size and color may cause the legibility of the error message to degrade.
- ❑ *Security problems caused by bad error handlers*: Sometimes error messages pose serious security risks by exposing sensitive data like port numbers, line number of internal code, type of server and internal configuration of systems. Mixing machine communicated errors and error-handling system may simplify the process of writing error messages but the risk of a security lapse runs high when such error messaging systems unintentionally channels out internal and sensitive data.

### **Examples of related bugs and other known issues**

Problem with "Hotwire.com": lack of a useful error message

<http://www.phototour.minneapolis.mn.us/essays/hotwire.html>

## **Human Error**

Though human judgement and perception is far more superior to any machine, the human tendency to err is always a risk. All shopping cart centric e-commerce systems involve some human action and intervention in the form of data entry, data upgrade, system upgrade, and system design. The chance of human error is equal on both the retailer's side and the user's side. The Common human errors in shopping cart are incorrect price entry and erroneous handling of back end processes. Below are some common risks that exist due to human errors

- **Human error on the retailer side**
  - ❑ Risk of price glitches: incorrect price entry, incorrect data feed, incorrect database configuration and all other forms of incorrect human data entry
  - ❑ Quantity glitches, incorrect entry of numeric inputs, input in wrong format.
  - ❑ System time incorrectly set, all time stamps on order placements out of sync
  - ❑ Administrator forgot to restart the web server or shut it down by mistake
  - ❑ Back-end human error: wrong item sent, or package inter-changed etc
  - ❑ Shopping cart configured incorrectly
  - ❑ Administrator erased custom settings by mistake
  - ❑ System reset to default by mistake

- ❑ Security breaches and system security compromises due to deliberate or non- deliberate human action
  - ❑ Forgot to backup the files
  - ❑ Corrupted the configuration file by mistake
  - ❑ Erased data or deleted files by mistake
  - ❑ Physical failures induced to the shopping cart system and its underlying hardware, due to bad handling, accidental damage caused by human action
  - ❑ Human error in entering the correct email address when sending confirmation of order placement (in non-automated systems)
  - ❑ Typos, grammatical mistakes, and incorrect language structure usage in content pages
  - ❑ Any large-scale human disaster or man-made disaster that causes physical damage to underlying e-commerce system.
  - ❑ Transaction aborted due to non intervention of required personnel
- **Human error on the customer side**
    - ❑ In-correct selections, in-correct navigation, in-correct understanding of the shopping process could be some top-level errors on the shopper's side.
    - ❑ Adding the wrong quantity, filling up information in the wrong fields, filling up incorrect information, specifying wrong shipping address are some of the other errors that cause the e-commerce system from delivering the items purchased through the shopping cart.
    - ❑ Entering the wrong data type, entering in the wrong format (ex. Date), selecting the wrong shipping options.
    - ❑ Deliberate or non- deliberate abortion of the transaction process.
    - ❑ Loss of shopping cart state and subsequent abandonment of shopping cart due to erroneously closing the browser.
    - ❑ Entering wrong credit card number or selecting wrong credit card type or entering the expiration date in the wrong format or order.
    - ❑ In-correct usage of the shopping cart functionality, like pressing the confirmation button multiple times, clicking on selection buttons multiple times causing errors in the order placement.
    - ❑ Trying to access the shopping cart in an incompatible underlying environment or using an older incompatible version of the browser, or having scripts and cookies disabled.
    - ❑ Do not have/ has not installed the required plug-ins or media software that is required to view the shopping cart catalog.
    - ❑ Wrong shipping methods requested for Alaska, Hawaii, Puerto Rico, and international addresses. Only UPS Second Air, FedEx 2Day, and USPS Priority Mail deliver to these addresses.
    - ❑ The Billing and Shipping addresses are reversed
    - ❑ Wrong e-mail address entered

### **Examples of related bugs and other known issues**



**Ashford.com flaw allows "free" purchases**

<http://news.com.com/2100-1017-233806.html?legacy=cnet>

**IBM customers buy \$1 laptops in site snafu**

<http://news.com.com/2100-1017-235771.html?legacy=cnet>

**Pricing mistake prompts Buy.com rush**

<http://news.com.com/2100-1017-221397.html?legacy=cnet>

**AOL nightmare: ordered a digital camera from AOL, received McAfee Office 2000 instead!!**

[http://www.zdnet.com/anchordesk/talkback/talkback\\_229502.html](http://www.zdnet.com/anchordesk/talkback/talkback_229502.html)

**United to honor dirt-cheap online ticket fares**

<http://www.itworld.com/Tech/2409/CWSTO57853/>

**Amazon.com hit with pricing glitch**

<http://www.computerworld.com/industrytopics/retail/story/0,10801,47949,00.html>

**Attache cases go for a penny apiece after pricing glitch at Staples.com**

<http://www.computerworld.com/industrytopics/retail/story/0,10801,57891,00.html>

**Customer outrage prompts Amazon to change price-testing policy**

<http://www.computerworld.com/industrytopics/retail/story/0,10801,50153,00.html>

**Amazon charging different prices on some DVDs**

<http://www.computerworld.com/managementtopics/ebusiness/story/0,10801,49569,00.html>

**The Price Isn't Right: A keying mistake set the price at a mere \$26.89 instead of \$299.**

<http://thestandard.net/article/0,1902,24690,00.html>

**Amazon glitch spurs shopping spree**

<http://www.usaicorp.com/cc/clips/2000/080200/amazonglitch.htm>

**Coding glitches main culprit in e-tail fire sales**

<http://news.com.com/2100-1017-244280.html?legacy=cnet>

**Price goofs in e-commerce**

[http://www.augustachronicle.com/stories/053101/fea\\_124-3979.shtml](http://www.augustachronicle.com/stories/053101/fea_124-3979.shtml)

**Risks due to *Calculation/Computation* errors**

A shopping cart has various calculations and computations like discount calculations, billing calculations, shipping and handling calculations and tax calculations. Summarized below are some common risks due to *calculation and computation errors* that cause shopping carts to fail.

- **Discounts/Coupons and special offer calculations**
  - ❑ Coupons in the online world are generally a set of numbers that accord a pre-mentioned discount. Common errors are Incorrect sequences of numbers, a mistakenly swapped set of numbers (denoting a different product and different discount)
  - ❑ Coupons being accepted by the system, after its expiration date.
  - ❑ An infamous bug has been that of, allowing the same customer to use the coupon multiple times until the total price has been completely discounted.
  - ❑ Coupon functions OK, but the billing system does not honor the coupon code and continues to charge the full and non-discounted price.
  - ❑ Coupons with conditions are also error-prone, because sometimes the conditions that make the coupon valid have errors and make the coupon deemable under all conditions!
  - ❑ Some coupon codes that are still under the development stage and not yet been open for public view get “crawled” by search crawlers. And the public gets away with some free shopping!
  - ❑ One other bug mentioned in the bug list below highlights how a user could not place a discount over a credit voucher she had, because the system could process either a discount or credit but cannot process both together!
  - ❑ Errors in “Quantity available” or “in stock value” displayed in the catalog. This may be due to incorrect computation of inventory stock value. Risk is there will be a delay in shipping the order or the order may never be delivered.
  - ❑ Check for all discount options. Discount by total percentage may work but discount by total weight may not work.
  - ❑ Check for issues caused by duplicate items with quantity discounts.
  
- **Pre-checkout/Check-out calculations**
  - ❑ Some shopping carts show the total only after checkout, but show incorrect and incomplete calculations when checked in the pre-checkout stage.
  - ❑ Some shopping carts display only the price of the item in the pre-checkout stage but omit all other additional costs like shipping, taxation and tend to show a large cumulative price when user is checking out,
  - ❑ Hidden costs not shown in the pre-checkout stage
  - ❑ Multiplication errors when multiplying prices in real numbers with integer quantities and subsequent error in displaying the total price.
  - ❑ Decimalization errors in checkout figures
  - ❑ Errors may occur in currency conversion when more than one type of currency is accepted and also subsequently conversion rate tables may be inaccessible or maybe outdated
  
- **Taxation calculations**
  - ❑ Taxes are applied only to the items, but like one of the bug examples quoted in this section, sales tax was applied to shipping costs!

- ❑ Even when an order has been placed for multiple copies of the same item, sometimes tax may be mistakenly applied only to one copy due to the system's internal calculation code.
  - ❑ When the issue of state taxes is encountered, common problems include wrong application of 'state' factor to the tax calculations, mix-up in estimating taxes for the region of delivery.
- **Shipping calculations**
  - ❑ Some sites which interface with other sites such as UPS to calculate 'Shipping costs' skip calculations on shipping when the other site is down and may not provide accurate total costs.
  - ❑ Again sites that interface with external sites for accessing their shipping tables, may incorrectly compute shipping costs when large quantities of heavy items are ordered, this error occurs due to the Maximum limit on the weight that is available in the cost calculator tables. For Ex. The UPS site allows calculation of shipping costs for packages with a maximum total weight of 100 LBS. Any order above that weight may be miscalculated since the calculation will continue to be based on 100 LBS
  - ❑ International shipping is another error-ridden area, since most international air shipping costs keep changing and also vary with respect to destination, generally the risk remains that the table used for calculation is outdated.
  - ❑ Most carts calculate on the basis of price threshold, weight threshold, quantity threshold, line item threshold and sometimes no charge. Risk is high that error may occur due to erroneously using the wrong model for calculation. A single very heavy shipment may then cost very less, if quantity is mistakenly used for the purpose of calculation.

## **Examples of related bugs and other known issues**

### **Spring forward leaves eBay behind**

<http://www.computerworld.com/industrytopics/retail/story/0,10801,59222,00.html>

### **Glitches let net shoppers grab free goods**

#### **Botched coupon deals let shoppers waltz out with free or nearly free deals**

<http://news.com.com/2100-1017-242811.html?legacy=cnet>

### **Macys.com says no to unauthorized coupon codes**

<http://ciscomp.com.com/2100-1017-251548.html?legacy=cnet>

### **Shoppers seize unauthorized discounts at Macys.com**

<http://ciscomp.com.com/2100-1017-251334.html?legacy=cnet>

### **AltaVista credits players \$1 after contest error**

<http://news.com.com/2100-1017-242970.html?tag=mainstry>

### **Staples.com nailed again by its own Net coupons**

<http://ciscomp.com.com/2100-1017-244220.html?legacy=cnet>

## Online shopping, glitches and gotchas

<http://www.nwfusion.com/newsletters/techexec/2001/01156806.html>

### Risks due to *Software Upgrade* errors

Due to the dynamic nature of their content, web stores and shopping carts undergo frequent updates, upgrades and changes. But these frequent changes tend to frequently break things and cause havoc when the site opens up for business after the upgrade. Listed below are some of the risks posed by software upgrade in shopping carts and e-commerce systems

- **Software upgrade on the server side**
  - ❑ A common error is the failure to backup the web-store before Upgrade.
  - ❑ Accidentally over-writing the product database file during upgrade
  - ❑ Non-removal of staging files before upgrade may lead to corruption of the shopping cart
  - ❑ Failure to update or reset correct file permissions in the shopping cart after upgrade process, this causes some pages to show “Unauthorized to view” errors when the user clicks on a catalog page
  - ❑ Many software upgrade processes look for folders with standard names. For example, CGI based shopping carts look for standard CGI directory path. Any deviations from the standards pose the risk of an incomplete install/upgrade
  - ❑ Some upgrades corrupt the shopping cart by changing the default file types to newer file types. And this newer file type may not be compatible with clients that use it.
  - ❑ Files upgraded successfully but did not to make changes go ” live" after upgrade!
  - ❑ Failure to check the OS compliance of host server before the upgrade
  - ❑ Failure to verify the host server's software and hardware requirements before upgrade
  - ❑ Insufficient disk space available for the shopping cart upgrade process and the upgrade stalls before completion
  - ❑ Failure to update older and outdated content, before an upgrade or site re-design
  - ❑ Risk of mistakenly listing outdated and discontinued products by over-writing new files with older ones.
  - ❑ “*We ran two programs at the same time that will not run together*”, Upgrades performed without checking inter-compatibility between existing or newer software processes within the system.
  - ❑ Post upgrade “internal glitch” have caused orders from being processed in shopping carts, they generally occur due to new but mismatched data feed installs, convoluted linking due to addition of new links within the shopping cycle, Older links not removed and new links installed without targets.

- ❑ Upgrades to some parts of the system, may cause selective failures in dependent or related sections of the system. A common issue has been upgrades to client information databases, causing User authentication failures due to lockouts and denial of access to login processes.
  - ❑ A fix to one bug causes another! A common problem in conventional software too. A good example of this type of risk is the example of DoubleClick Ad failure mentioned in a bug listed below.
  - ❑ A “newer look” or “fresh look” after an upgrade may not always mean an error free look for the site, “newer look” changes the GUI and functionality and this leads to newer problems both in terms of functionality, usability and technical glitches leading to blackouts.
  - ❑ Another important risk is the risk of security problems that are caused by poor installation and in-complete installation that results in some security features being turned off.
  - ❑ Software upgrades sometimes sets all options to ‘default’ automatically after the installation is complete and in this in turn may over-write any existing customized options This leads to change in e-commerce system behavior and settings.
- **Client side response to server side software upgrade**
  - ❑ Browser incompatible with the new upgraded server side shopping cart

### **Examples of related bugs and other known issues**

#### **Amazon Endures Third Holiday Outage**

<http://www.ecommercetimes.com/perl/story/5870.html>

#### **Webvan stalls on the way to Thanksgiving dinner**

<http://news.com.com/2100-1017-248798.html?legacy=cnet>

#### **E\*Trade users locked out of trading**

<http://news.com.com/2100-1017-221117.html?legacy=cnet>

#### **Problems hit E\*Trade for third day**

<http://news.com.com/2100-1017-221192.html?legacy=cnet&tag=rldnws>

#### **Software Glitch Affects DoubleClick's Domestic Clients**

[http://www.atnewyork.com/news/article.php/8471\\_441871](http://www.atnewyork.com/news/article.php/8471_441871)

#### **Walmart.com runs into glitches**

<http://news.com.com/2100-1017-249390.html?tag=prntfr>

**When Buy.com [redesigned](#) its Web site on April 26, it mistakenly listed between 4,000 and 7,000 discontinued laser discs for \$1.11 apiece**

<http://news.com.com/2100-1017-225527.html?tag=rn>

#### **Dangerous shop service if installed the right way**

<http://exploiter.virtualave.net/9904-exploits/hhp-WebShop.txt>

## Yahoo introduces email bug after attack

<http://news.com.com/2100-1023-236686.html?legacy=cnet>

## Document Confidentiality

How secure is the e-commerce site? Is it safe to give my credit card number? Can someone get my order details and my personal information?

Document confidentiality means protecting private information from being leaked to third parties [8]. Compromises on this issue lead to serious security related failures. This category deals with issues like credit card information leaks, order information leaks, account information leaks, etc.

Shopping carts with advanced features provide direct linking with credit card processing agencies using secure protocols such as SSL or SET (Secure Electronic Transaction). But we need to remember that these secure protocols are also prone to failures and attack by malicious elements and can hence cause sensitive data loss.

Cryptography is a key technology that is used for protecting the system against such leaks and testers testing shopping carts that encrypt data in bill payment transactions need to know some simple ways in which cryptography fails?

- ❑ The risk of the cryptography algorithm failing because it contains patterns from the plain text and the algorithm can be guessed.
- ❑ The risk of the decryption key being guessed and hence may succumb to attacks such as brute-force attack.
- ❑ Risk of using lower bit keys to encrypt data. Lower the number of bits the easier it is to crack the key. 128 bits and higher are considered safe.
- ❑ Loss or corruption of a private key
- ❑ A key is compromised but failure to replace or remove the compromised key.

Cryptography is a well-published topic and lots of information is available on the web on encryption. It may prove futile for testers to test every aspect of cryptography, due to the complexity of the subject. But a basic idea of the risks associated with violating a few basic rules such as safe key selection or failure to replace corrupted keys will help in validating the security in the transaction stage of the shopping cart.

Apart from encrypting the data transfers to protect information, “Document Confidentiality” also involves physically safe guarding files and documents that contain sensitive and confidential information.

Refer to the bug example “**Shopping Carts Expose Order Data**”, where a poorly installed shopping cart exposed the order ‘log’ file with names, addresses, credit card numbers in a world-readable format. People could search for these log files just by entering simple key words from any search engine!

- ❑ Risk of exposing directories that hold sensitive files and allowing an external user to access the directory or folders from the web!
- ❑ Risk of setting improper read and writes permissions to these files due to which any external user can access and modify these important files.

- ❑ Risk of mistakenly configuring the email list server to include sensitive customer information or attach confidential files in public email listings and postings.
- ❑ Script errors that let users edit their URL by changing a few visible parameters like order number and grant them access to other user's records!
- ❑ Poor configuration of shopping carts may cause an attacker to gain entry to classified information (refer to the examples for more details)
- ❑ Risk of unfixed bugs or new bugs in databases and server software may open up serious security holes (Refer to the example bugs). [www.bugnet.com](http://www.bugnet.com) hosts a long list of security bugs in this category.
- ❑ Check for issues like where the shipping section shows "billing Information" in non-secure customer, emails.
- ❑ Check for issues, where instead of just the last four digits, all numbers of the credit card are exposed.

## **Examples of related bugs and other known issues**

### **E-Commerce Fears? Good Reasons**

<http://www.wired.com/news/ebiz/0,1272,44690,00.html>

### **Shopping Carts Expose Order Data**

[http://www.internetnews.com/ec-news/article.php/4\\_102621](http://www.internetnews.com/ec-news/article.php/4_102621)

### **Shopping Carts exposing CC data**

<http://exploiter.virtualave.net/9904-exploits/cybercash.cc.txt>

<http://exploiter.virtualave.net/9904-exploits/perlshop.cc.txt>

<http://exploiter.virtualave.net/9904-exploits/shopping.cart.cc.data.txt>

### **Expert finds hole in shopping carts**

<http://zdnet.com.com/2100-11-514435.html>

### **HQ for Exposed Credit Numbers**

<http://www.wired.com/news/ebiz/0,1272,44613,FF.html>

### **Which? under fire over security scare**

<http://news.bbc.co.uk/1/hi/sci/tech/1402222.stm>

### **Qwest Glitch Exposes Customer Data**

<http://online.securityfocus.com/news/431>

### **O'Reilly Leaks Geeks' Info**

<http://online.securityfocus.com/news/408>

### **United Airlines**

**Frequent fliers who logged onto United Airlines'**

**Web site got a look at other people's Mileage**

**Plus account information for more than 12 hours**

<http://www.siliconvalley.com/docs/news/tech/072275.htm>

## Hacker Posts Credit Card Info

<http://www.wired.com/news/technology/0,1282,33539,00.html>

## Risks due to Memory Leaks

These are the bane of many a site, and not surprisingly also a major cause of many e-commerce site crashes and also the cause of deteriorating performance of many e-commerce sites. Shopping carts are complex systems that use scripting code, run on a server, need an underlying OS and also need a browser on the client side to function. Hence a memory leak that occurs in any of the above components can cause the shopping cart to fail indirectly due to memory leaks!

Just as this article on [Web testing in ExtremeTech](#) puts it, “Memory leaks can range from irritating to debilitating. A merely irritating memory leak might involve a component growing until it crashes and re-spawns. This is still bad for the site, as a crashed component forces everyone to wait until it restarts. The worst-case scenario is a component that uses up more and more system memory (or worse, kernel memory) without exiting, until the entire system finally grinds to a halt.”

The common issues discussed here is a subset of a larger set of memory leak issues and bugs that can be found (in more detail) at Labmice.net [9]  
(<http://www.labmice.net/troubleshooting/memoryleaks.htm>)

- **Issues due to memory leaks in scripting code**

- ❑ Some scripts have maximum static string size and violation of which causes memory leaks, hence shopping cart pages that use heavy and long scripts to add navigational or functional capabilities may violate these size constraints resulting in browser throwing up “*Out of Memory*” errors when the cart pages are viewed.
- ❑ Bugs in the Script DLLs may cause memory leaks under specific conditions such as when the limit on the number of loops within a script is exceeded.
- ❑ Some inbuilt functions such as ‘string format’ functions (as in VBScript) in common script languages have known memory leak problems and frequent use of these functions in a high volume web environment such as ASP enabled shopping cart page may lead to serious memory leak problems.
- ❑ Enough cannot be said about the negative effects of sloppy programming practices, which are the root cause to most memory leaks, and badly written scripts that lock up resources can be equally lethal in shopping cart pages as in anywhere else.
- ❑ Since shopping carts are chiefly database driven, resource locking is a big risk where a database record or a file becomes unavailable for prolonged



periods of time because a particular site component has an exclusive hold on it.

- ❑ Older version of constituent components in the scripting environment can also be a potential cause to memory leaks, for example using an older version of perl interpreter engine in a perl-based shopping cart.
- ❑ Some design level decisions can also save the e-commerce site from potential memory leaks, one of them is to adopt a modular scripting framework where independent classes can be tested for memory leaks.

- **Issues due to memory leaks in browsers**

- ❑ IE 5.0 and 5.01 have had memory leak problems when images were re-sized using DHTML (Dynamic Hyper-Text Markup Language) . Since it is common to use scripts and DHTML to format images in shopping cart catalogs, care should be taken not to trigger memory leaks. This can potentially freeze the cart page, and cause performance problems and loading errors.
- ❑ Some browser methods like “StartDownload” methods consume excessive memory and do not return them back to the system. Hence shopping carts that offer downloadable files, product, software code and also host large sized image files are in danger of potential risks of memory leaks, triggered by calling these risky browser methods.
- ❑ Browsers are known to load and access several libraries. And shopping carts that spawn duplicate product browsers, pop-ups, ad browsers multiply the total RAM consumption. Since these browsers have known memory leakage problems associated with their versions, they may cause the system to come to a cranking stop, freezing any potential transaction midway.
- ❑ Memory Leaks in IE 3.0x: These versions (3.0/3.01a/3.01b) have progressive memory leaks; one of its manifestations slows down the performance and response of the browser slowly over a period of time. And thus media rich or heavy data content shopping cart pages faces serious performance issues due to these types of memory leaks.
- ❑ Another famous memory leak bug in IE 3.x is it caches page information when using the <Form Method=POST> tag and fails to free the memory until the application is shut down, this is a serious problem since many carts use GET and POST methods in their forms.

- **Issues due to memory leaks in underlying operating system**

- ❑ Undeleted threads are a major source of memory leaks in Operating systems, so if the underlying OS of an e-commerce system starts leaking memory due to undeleted threads then the hosted shopping cart begins to fail due to lack of memory available for its functioning
- ❑ Any third-party process that may be running on the operating system may cause unreleased memory, which may indirectly cause the shopping cart to fail due to underlying OS failure because of lack of sufficient memory.
- ❑ Some standard system libraries in older versions of operating systems may leak memory, so risk of not upgrading to newer underlying operating

systems in e-commerce sites transforms itself into a risk leading to memory leak.

- ❑ Also shopping cart sites, which have user written server side plugins created by server programming APIs, the user code may introduce serious memory leaks.

- **Issues due to memory leaks in server**

- ❑ Incorrect use of multithreading in Web server software can be a problem and may end up in a memory leak, since a shopping cart sits on a web server, a web server failure leads to a shopping cart being rendered non-operative.
- ❑ According to the different IIS FAQ lists and MS knowledge base articles, Inetinfo process in Internet Information Server (IIS) may leak memory when using SSL.
- ❑ Some web servers hosting software downloads or shopping carts offering large documents, eBooks, Media files as downloadable products have a high risk of suffering from memory leaks the equal to the size of the file being uploaded/downloaded if transfer is aborted prematurely. The cause may be due to some methods like *Request.BinaryRead* being called by ASP or other similar scripts.
- ❑ When trying to access member accounts in a shopping cart, small memory leaks may occur when a lookup of the current domain name is performed. A pointer to the domain name may be saved in a global location without freeing the previous domain name already stored there.

## **Examples of related bugs and other known issues**

### **IIS Memory Leaks**

<http://www.iisfaq.com/MemoryLeaks/>

### **Memory Leaks in OS**

<http://www.labmice.net/troubleshooting/memoryleaks.htm>

### **More on Resource Leaks**

<http://www.willows.com/listarchives/dev/twindev-1998-jul/0155.html>

## **Risks due to *Insufficient Capacity Planning***

According to this Microsoft TechNet paper [10] “Capacity planning is the process of measuring a Web site's ability to serve content to its visitors at an acceptable speed. This is done by measuring the number of visitors the site currently receives and by how much demand each user places on the server, and then calculating the computing resources (CPU, RAM, disk space, and network bandwidth) that are necessary to support current and future usage levels”. The idea of categorizing the risks into the following three categories is derived from the same paper.

- **Risks based on the number of users and usage**
  - ❑ Shopping cart performance degrades due to increase in site users disproportionate to existing capacity.
  - ❑ No Increase in the number of users but increase in the activity of the users, increase in terms of catalog page hits, latency time, increase in usage of search activity, increases in shopping cart update cycles. The increase in such heavy resource consuming activities may upset the capacity planning equation, which maybe based on the number of users and not usage.
  - ❑ The most common cause that leads to sudden load and causes deficiency in system capacity are the seasonal increases in customers especially the “Holiday shoppers”. Test shopping cart for performance and scalability under realistic loads
  - ❑ Increase in the number of transactions involving third party components like Billing cycles, Credit card authorizations and account transfers, where the insufficiency in the capacity of the third party systems will indirectly cause the shopping cart and the e-commerce site to stall.
  - ❑ Resource consumption also depends on the stage of the shopping cycle. For Example: The checkout stage uses more pages, more CPU, more DB transaction cycles and more server utilization than the catalog ‘browse’ stage. One has to always plan for sufficient capacity and availability for all stages of the shopping cycle keeping in mind the changes in the requirements at each stage.
- **Risks based on computing Infrastructure**
  - ❑ CPU Insufficiency may be a big risk if there is an excessive demand placed on CPU by the Web Server or the Database Server. Web servers especially tend to consume more CPU cycles than the corresponding Database server in the system.
  - ❑ If the shopping cart spawns a new process every time a user invokes it, and no mechanism exists to limit the maximum number of shoppers then very soon the processes will choke the available CPU and cause the entire system to slowly crash.
  - ❑ An operation may cost less in terms of resource consumption, but if the frequency of that operation is high, then very soon we will have a capacity insufficiency risk. Generally product pages and search pages are moderate in their cost but search page operation is very frequent due to which very soon, it may become the largest resource consumer despite its cost being less.
  - ❑ Also Shopper capacity is determined by the underlying Operating Systems. For example According to this paper [10] Win NT reaches CPU utilization of 96.40% at shopper’s load of 1000 while Win 2000 reaches CPU utilization of 72.89% at shopper’s load of 1000.
  - ❑ If any shopping cart operation like basket load or catalog load is memory intensive, then the underlying web server may run into memory deficit very soon.

- ❑ If any shopping cart operation forces the web server's page-able process to page to disk, it is bad news and will affect the performance of the web server badly.
- **Risks based on site content complexity**
  - ❑ Network capacity may become a bottleneck if shopping cart uses high static content like large images and static HTML.
  - ❑ Poor site design where heavy elements and heavy content pages are called more often than the lighter ones due to which the resource consumption gets unevenly distributed and resource consumption becomes very high. Higher demand should have 'light' content and we can afford to make pages with lesser demand 'heavy'.
  - ❑ Advertisements retrieved from ad databases, Customizations to fit shopper's choice, ActiveX control driven Menus and Java based menus are some of the complex site components that potentially affect the capacity of the system and tend to pose a risk to the functioning of the shopping cart.

## **Examples of related bugs and other known issues**

### **Crashing Success for the Web?**

#### **For Online Retailers, a Make-or-Break Year Could Find Sites Overloaded**

<http://abcnews.go.com/sections/business/TheStreet/onlineretail991202.html>

### **Charles Schwab Web Site Crashes**

<http://www.binarythoughts.com/article.cfm?StoryID=237>

### **Encyclopaedia Britannica's New Web Site Crashes**

[http://www.infowar.com/p\\_and\\_s/99/p\\_n\\_s\\_102299e\\_j.shtml](http://www.infowar.com/p_and_s/99/p_n_s_102299e_j.shtml)

### **E-tail sites crash over holiday weekend**

<http://news.com.com/2100-1017-249048.html?legacy=cnet>

### **Webvan running out of Thanksgiving goodies (may also go into process failure)**

<http://news.com.com/2100-1017-248881.html?legacy=cnet>

### **Customers locked out of Virgin Megastore's online sale**

<http://news.com.com/2100-1040-230643.html?legacy=cnet>

## **Accessibility Risks in Shopping Carts**

According to statistics provided at [www.webaim.org](http://www.webaim.org) [11] "an estimated 20 percent of the population in the United States (40.8 million individuals) have some kind of disability, and 10 percent (27.3 million individuals) have *severe* disability. The 27.3 million individuals with severe disabilities are limited in the way that they can use the Internet". W3C's Web Accessibility Initiative (WAI) has produced the [Web Content](#)

[Accessibility Guidelines 1.0](#) [12], which explain in detail how to make a Web site accessible for people with a variety of disabilities.

Detailed lists of risks that affect shopping carts in terms of accessibility failure are provided below. They have been categorized with respect to the different severe disabilities that affect Internet users and more specifically people who use shopping cart enabled e-commerce sites.

## ❖ **Visual Impairments**

- **Blindness**

- ***In General***

- If the cart catalog is categorized under headings and captions to denote product categories, shipping options, billing options, test the content with a screen reader and check if the document's text or other text equivalents make sense? Test and verify that the change in categorization is understandable to a person who cannot see the headings or captions.
- If you are using style-sheets to render your shopping cart pages, check if it is possible for screen readers and non-CSS supporting browsers to render the cart page correctly.
- When dynamically updating shopping cart pages, the equivalents to dynamic content may not get updated when the products, rates, or prices are dynamically updated. Thus only non-updated data would be available to impaired or disabled users.
- If you are using an image-mapped-shopping cart page, check for redundant links to every navigable section.
- In shopping cart product pages, where size tables, price tables, shipping tables, schedule tables are used; Test if row and column headers are identified. Otherwise the screen reader will output a stream of non-distinguishable data.
- Also when testing tables that use two or more logical levels, Check if alternative text exists to identify which cell is a header cell and which contains data.
- When frames or layers are used to create the browse catalog in the cart, check each frame for title. This will help identification and navigation when read by screen readers.
- Some of the tools that help disabled or impaired users do not support programmable objects such as applets or scripts. Check for alternative functionality in the shopping cart under a situation when the user has these options turned off
- Use of pop-up shopping carts, pop-up advertisements, pop-up alerts, sales pitches in new windows takes the control off the active window that's being read and confuses the

reader by making the screen reading software alternate between the different windows!

- ❑ Navigation across the shopping cart pages should be consistent and straightforward, Look for looping navigation and random return-backs and other similar issues that cause the screen reader navigation problems.

- **Text Equivalent**

- ❑ Check for equivalent test for every non-text element in the product catalog of the shopping cart
- ❑ Check for issues where the text equivalent describes the graphic but not the content in the graphic image!
- ❑ Look out for open or broken ALT tag, this disables the screen reader from reading the content inside the tag.
- ❑ If tables are used to format the appearance of equivalent text, test using a screen reader whether the content read from the table makes sense. Tables tend to confuse the screen reader and the screen readers tend to read the content in different cells in a haphazard way.
- ❑ Check if by error the alternative text equivalent is provided in a language not same as the language of content. Apply this test to all sites that have multi-lingual international sites.
- ❑ Check for typos, spelling errors and word jumble in the alternative text. Screen readers do not have the ability or the intelligence to notice the errors and it reads out non-tangible words to the user.
- ❑ Check for unknown abbreviations, acronyms, and unfamiliar complex words and jargon describing non-text elements in the cart
- ❑ Check if the items that have been added into the shopping cart have alternate text, which the screen reader can read out so that the user can verify that the item she just purchased, has been added.
- ❑ Check if the images of credit cards that are displayed at checkout have alternative text. Otherwise, the user will not be able to make out what cards are accepted.
- ❑ Test shipping cost calculators, gift-wrapping cost calculators and other such user-aid tools for equivalent text support. Test each button and field, for alternative text describing the functionality.
- ❑ In the billing and shipping section of a shopping cart, test the order of entry and order of tabs with a screen reader. Sometimes screen readers read tabs in the wrong order.
- ❑ Test multimedia presentations and alternative text for correctness of data and also check if they appear in sync.

- **Color-Blindness**

- **Total Color Blindness**

- Testers should test if an alternative text exists for text that conveys information by means of color. For example, if all items marked for sale are marked in 'red' or new items are marked in 'blue', then equivalent alternatives should exist in regular black text.
    - Testers should test for alternatives when color is used as the primary way to indicate an action, say links turns purple from blue, when it is clicked. Similarly some shopping carts mark 'visited' categories by changing its color when a user clicks on it. This color change is to aid the shopper in keeping track of what they have seen and what they have not seen? But a person with color blindness may not notice the change in color as he has trouble identifying the colors. An alternative may be using an object like asterix or a cursor instead of color to identify change in state.

- **Color deficiencies**

- People with a color deficiency can see some colors but some pairs of colors look the same to them. So, for example, foreground and background colors may appear the same. Look for difficult color combinations in catalog design, link identification, announcements and so on. Reds, greens, oranges and yellows are the most likely to cause problems. . For color combinations, see <http://www.webaim.org/intro/intro>

- **Low Vision**

- People with low vision impairment use screen enlarger software to increase readability of small text but the enlarger limits the visible area of the browser screen. Screen designs that communicate well at normal text size may be confusing and hard to use when viewed in an enlarged mode.
    - If pages rely on scrolling, screen enlargers will yield pages that require more scrolling. If items on the screen are not appropriately grouped, the user will have to remember and correlate too many details that should be displayed together
    - Another risk lies in the use of graphics with embedded text for product catalogs, because due to enlargement of screen space, the images may get highly pixilated and embedded text may become hard to read.
    - People with age-related visual impairments—such as macular degeneration, glaucoma and cataracts prefer

shopping cart pages and functionality that can be enlarged, scrolled and purely textual devoid of much use of graphics.

## ❖ **Hearing Impairments**

### ○ **Deafness**

#### □ **Lack of visible textual support**

In sites where sounds signifies a buying process, purchase alerts, error message, instant messaging from a live representative, then testers should look for alternative visible textual support.

#### □ **Test for the absence of subtitles or other text**

Sub titles or alternative synchronized text, should supplement video catalogs, virtual demos of products product information sessions and help videos for shopping.

### ○ **Deaf-Blindness**

#### □ **Test for special cases of double disability**

Test for all risks that are applicable to Blindness, but since Deaf-Blind use screen readers that convert text into Braille (through a refreshable Braille device attached to the computer), you have to test to confirm that there is text that is readable by the Braille reader. Sound as an alternative to visible text is workable for the blind but will not work for people with double disabilities, such as deafness and blindness.

## ❖ **Mobility impairments**

□ Mobility impairments range from minor to major problems restricting voluntary muscle movements. Because of lack of dexterity, the user may find it difficult to click on small (single letters and alphabets), product links and catalog navigation links. Similarly, look for small functional buttons or other tiny targets that must be hit precisely.

□ Others using devices to access the keyboard, such as a mouth stick or a head wand have lower dexterity than regular users, so test auto refresh, time outs and auto-exits from secure billing areas and verify that the time available is sufficient for them.

□ Users using devices such as head wand have to shake their head 20 times to browse through 20 links! Hence test if the unnecessary links can be minimized, or short cuts to skim past groups of related links can be offered?

□ Keyboard shortcuts and keyboard functionality is vital to users with mobility impairments; risk is high that they will be neglected if too much mouse-centric functionality is added.



❖ **Cognitive impairments**

- ❑ Some of the better-known cognitive impairments are the Downs Syndrome, Alzheimer's disease and the lesser-known cognitive impairments include reading and learning disorders. Individuals with cognitive impairments often benefit from graphics or icons that supplement the text, providing a monotonous small sized 'text only' interface devoid of any meaningful graphics and animations may not benefit such users much.

❖ **Seizure disorders**

- ❑ Beware of flickering sales tickers, ad banners, notifications, alerts, or interactive messages that are provided in some shopping carts. These seemingly harmless gizmos may be a serious risk to people with seizure disorders if their frequency of flickering is between 2 Hz to 55 Hz.

## Related Internet Links

1. Western Australian Electronic Commerce Center, Information on ***Accessibility and Usability***  
(<http://www.ecommercecentre.online.wa.gov.au/matrix/acc.htm>)
2. **Section 508:** The amended Rehabilitation Act that requires Federal agencies to make their electronic and information technology accessible to people with disabilities.  
(<http://www.section508.gov>)

## Usability Risks in Shopping Carts

As per the definition provided by usability.gov, Usability is the measure of the quality of a user's experience when interacting with a product or system — whether a Web site, a software application, mobile technology, or any user-operated device.

And hence a general agreement is that a “Usable Web interface is one that is

- ❑ Easy to learn
- ❑ Easy to use,
- ❑ Culturally appropriate
- ❑ Not discordant with the user's expectations of how the program should operate,
- ❑ Responsive (fast enough responses that the user doesn't notice waiting for the program to catch up).

Provided below is a list of issues that may seriously hamper the usability of a shopping cart in an e-commerce site. Real life bugs support some of the risks and issues and some others have Do's and Don'ts on how to avoid the risk.

❖ **Pop-Ups**

- ❑ Do not use pop-up window shopping carts because If the user clicks elsewhere in the main window, the pop-up is sent "behind" the main window
- ❑ Test shopping carts with pop-up/Ad eliminating software turned ON. Pop-up shopping carts may not work if the pop-up eliminator is ON.
- ❑ Check if Pop-up shopping carts have sufficient "real-estate" space when the user adds more items.

❖ **Functionality**

- ❑ Provide "remove item" or "add item" buttons instead of asking the user to change "item quantity number"- it is easier and more error free.
- ❑ Look for items that have not been linked back to the “item”/catalog page.
- ❑ Check if the “number of items” in the cart is displayed. Users prefer carts that show the current data and state, like how many items are in the cart? What is the total? Etc
- ❑ Check if the "Continue Shopping" and "Proceed to Checkout" buttons are visible.

❖ **Navigation**

- ❑ Look for unnecessary steps between item selection and checkout. The more clicks, the more confusion and the greater the probability that the customer will abandon the transaction.
- ❑ Do not link to any external site/page from the shopping cart page- leads to shopper getting confused/uninterested-leading to shopping cart abandonment.
- ❑ Check if the shopper is able to navigate back to shopping process, after "adding" or "removing" items
- ❑ Check if it is possible to add additional items directly from the cart page, instead of going back to browsed pages. This improves functionality and enhances usability.
- ❑ If providing detailed info on products to users, then test if you are able to return back to the shopping cart from the detailed page and also check if the state of the shopping cart is maintained.

❖ **Ease Of Use**

- ❑ Check if thumbnail photos of the items can be added to the shopping carts in addition to a text description, this re-assures the customer that the right item has been added to the shopping cart
- ❑ Try enhancing the usability by providing an auto-update cart facility after user has added/removed item.
- ❑ Check for appropriate positioning of buttons. Place "Continue Shopping" on the left and "Checkout" button the right, users perceive it analogous to "back" and "going forward" respectively
- ❑ Presence of standard "credit card" images on the UI adds trust psychologically on the site's security. Check the shopping cart for images or text that might cause mistrust in the user.
- ❑ Check the if the UI provides functionality for discounts and coupons. Provide separate field in the UI to display discounts due to coupons, it helps user note the discounts better
- ❑ Provide separate columns to display "total" bill as the user adds items to the cart
- ❑ Too much information to type into the cart - this common problem should be avoided
- ❑ Do not limit the features of the shopping cart--keep it flexible
- ❑ Cart is too hard to use -reduce functional complexity
- ❑ Check if the user is conveyed the information of order placement. Warn the customer when the transaction becomes final; do not surprise them by abruptly billing their contents.
- ❑ Check forms against data requirement. Collect only essential information about the user that is absolutely a must for completing the deal, unnecessary questions and making optional questions compulsory makes the user experience bad.
- ❑ Check for Hi-Tech whiz creations like flash display of catalog and constantly flashing blue lights in a shopping cart because it may reduce the usability of the cart. A classic example of a site that got booed away due to its technical gimmickry was [www.boo.com](http://www.boo.com)
- ❑ Check for plug-ins or media files that are not common in any general browser software, and recommend not using them. Expecting users to download software to shop at your site is high handedness! And may cost you heavily in terms of loss of customers to other competitors.
- ❑ Provide the user with the functionality to choose the mode of shipment, Check for fixed default radio buttons, non-flexible shipping options, erratic placement of multiple selection checkboxes

- ❑ Not sticking to known paths in navigability and sequence of shopping decreases the usability of the shopping cart, Check for odd sequencing issues like re-sequencing shipping costs after the user has been billed and charged. This will confuse the customer about whether the purchase was executed or not!
- ❑ When new functionality is added to the shopping cart- Check if it is user-understandable otherwise provide help
- ❑ Check for odd naming of known metaphors.
- ❑ Check if shipping can be calculated before checkout. Shoppers prefer getting an idea of the total cost of the item.

Eric Myers's Case study [13] on shopping cart experience and Andrew Starling's E-commerce and Usability [14] researches some of the above-mentioned risks in a detailed manner.

### **Examples of related bugs and other known issues**

#### **I hate waiting**

[http://www.zdnet.com/anchordesk/talkback/talkback\\_229493.html](http://www.zdnet.com/anchordesk/talkback/talkback_229493.html)

#### **You forgot 'overcomplicated technology'!**

[http://www.zdnet.com/anchordesk/talkback/talkback\\_229472.html](http://www.zdnet.com/anchordesk/talkback/talkback_229472.html)

#### **Dell Computer's site has got to be the worst for hiding charges**

[http://www.zdnet.com/anchordesk/talkback/talkback\\_229509.html](http://www.zdnet.com/anchordesk/talkback/talkback_229509.html)

#### **Last minute surprises make me crazy too**

[http://www.zdnet.com/anchordesk/talkback/talkback\\_229513.html](http://www.zdnet.com/anchordesk/talkback/talkback_229513.html)

#### **Cookie crumbled**

[http://www.zdnet.com/anchordesk/talkback/talkback\\_229516.html](http://www.zdnet.com/anchordesk/talkback/talkback_229516.html)

#### **Convert Browsers to Buyers (Article)**

<http://builder.cnet.com/webbuilding/pages/Graphics/Ecommerce/index.html>

#### **E-commerce and Usability**

<http://wdvl.internet.com/Authoring/Design/Basics/ecom1.html>

#### **Two usability bug examples: Usability Testing**

[http://www.smartisans.com/usability\\_testing.htm](http://www.smartisans.com/usability_testing.htm)

## Appendix: Terminology

- ❑ **Test idea:**

A test idea is a brief statement that identifies a test that might be useful. A test idea differs from a test case, in that the test idea contains no specification of the test workings, only the essence of the idea behind the test. Test ideas are generators for test cases: potential test cases are derived from a test ideas list. A key question for the tester or test analyst is which ones are the ones worth trying. (We thank Brian Marick for introducing us to this term, and the level of analysis that comes with it.)

- ❑ **Risk category**  
We call the 60 top-level categories that constitute the outline, risk categories. Each of them identifies a class of problems that might arise in the function under test.
- ❑ **FMEA**  
Failure Modes and Effects Analysis is a common method of risk analysis for complex or life critical systems. The process involves identifying the failure characteristics of individual components and determining the effects of those failures as they propagate across the systems in which they are embedded.
- ❑ **Failure mode**  
By definition, “Failure Modes are sometimes described as categories of failure. A potential Failure Mode describes the way in which a product or process could fail to perform its desired function (design intent or performance requirements) as described by the needs, wants, and expectations of the internal and external Customers”[2]
- ❑ **Software bug taxonomy**  
Taxonomy is the theoretical study of classification including its bases, principles, procedures and rules [Grolier Incorporated 1993] and classifications that are created by categorizing the different software bugs are called bug taxonomies.
- ❑ **Risk-based testing**  
Risk-based testing is a strategy for selecting test ideas and designing tests based on the types of problems you imagine might be present in the product. Risk-based test management is a prioritization scheme--the test management team decides which types of tests or potential problems to invest testing time in based on perceived (or estimated) probability of failure and expected damage that will result if the product actually fails in this way.

## References:

- [1] A. Biraj Rath, NIIC (USA), Inc., P. M. Raj Nath, Microsoft Corporation, P. M. Mukesh Agarwal, Microsoft Corporation, P. M. Jas Lamba, NIIC (USA), Inc., R. George Gianopoulos, Microsoft Corporation, and C. Laura Hargrave, Microsoft Corporation, "Testing Process," July 2000.

- <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnentdevgen/html/testproc.asp>
- [2] FMECA, "FMEA Methodology."  
<http://www.fmea.com/ffmethod/definiti.htm>
- [3] S. Amland, "Risk-Based Testing and Metrics," *EuroSTAR 99, Barcelona, Spain*, 1999.  
[www.amland.no/PowerPoint/EuroSTAR99Presentation.ppt](http://www.amland.no/PowerPoint/EuroSTAR99Presentation.ppt)
- [4] I. 9126, "Software engineering -- Product quality -- Part 1: Quality model," *International Organization for Standardization*.
- [5] P. Martin, "Managing Database Server Performance within an Electronic Commerce Framework."  
<http://www.cs.queensu.ca/home/cords/publications/m99.pdf>
- [6] Oracle Corporation, "Oracle9i Database Administration: Recover Databases."
- [7] C. Dille, "Known HTTP Proxy/Caching Problems," May 14, 2001.  
<http://www.wrec.org/Drafts/draft-ietf-wrec-known-prob-03.txt>
- [8] L. D. Stein, *Web Security: A step-by-step reference guide*, May-2000.
- [9] Labmice.net, "Memory Leaks."  
<http://www.labmice.net/troubleshooting/memoryleaks.htm>
- [10] L. Louis de Klerk (Inobits Consulting Pty., Jason Bender (MSNBC), "Microsoft Enterprise Services White Paper E-Commerce Technical Readiness," *Microsoft TechNet*, April 2000 version 1.0.  
<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/e-commerce/maintain/optimize/capplan.asp>
- [11] WebAIM, "Introduction to Web Accessibility," 2002.  
<http://www.webaim.org/intro/intro>
- [12] W3C-WAI, "Web Content Accessibility Guidelines 1.0," 5-May-1999.  
<http://www.w3.org/TR/WCAG10/>
- [13] E. G. Myers, "Case Study #4: The Shopping Cart Experience," 2001.  
<http://www.egmstrategy.com/carts4.html>
- [14] A. Starling, "E-commerce and Usability," November 5, 2001.  
<http://wdvl.internet.com/Authoring/Design/Basics/ecom1.html>