National Journal of Multidisciplinary Research and Development
ISSN: 2455-9040
Impact Factor: RJIF 5.22
www.nationaljournals.com
Volume 3; Issue 1; January 2018; Page No. 435-445

Generating bug taxonomies for testing

[1] Arvind Sharma, [2] Sanjay Babu Sharma
[1] Assistant Professor, Department of Computer Science & Engineering, MITRC, RTU Kota Rajasthan, India
[2] Research Scholar, Department of Computer Science, Singhania University Pacheri Bari Jhunjhunu, Rajasthan, India

Abstract
Taxonomies have been created and used widely from physical sciences to physical anthropology.
In the business world we see a lot of talk on Enterprise Taxonomies and Business Taxonomies. In psychology we hear about Personality Taxonomies, Gesture Taxonomies and Krathwohl's Affective Taxonomy, Educational psychologists use what is famous as Bloom's Taxonomy of Educational Objectives (1956). Many fields have used classification systems and taxonomies, but their applications and requirements have been different.
We have seen the word taxonomy also being used to describe the thesaurus or classification scheme, to organize information on the web.

Keywords: enterprise taxonomies, business taxonomies, physical sciences

Introduction
Taxonomies have found wide applicability in areas of computer science wherever an organized and systematic approach of organizing is needed. For example, instances of the usage of the word
'taxonomy' in computer science can be found in an array of topics such as Taxonomy of human- computer Interactions, Taxonomy of computer system architectures, Taxonomy of computer inputs, Taxonomical classification of meta-data, Taxonomies in controlled vocabularies, etc.
Some taxonomies are system specific (i.e. eccentric -- suitable for only one environment and application) and few others are generic and can be applied generally to a wide range of systems. Error taxonomies classify types of errors and error mechanisms. Learning taxonomies deal with required behaviors and types of learning. Functional taxonomies look at system functions.
Vulnerability taxonomy, Incident taxonomy, Attack taxonomy, deals with the classification of security bugs.
Among all sub-specializations within computer science, computer security and vulnerability analysts have probably employed taxonomies in the largest way to classify security holes, vulnerabilities and other related security breaches.
In this paper, for the sake of brevity, I have listed sample taxonomies from just three categories.
1. A few examples of general fault taxonomies and their objectives.
2. Examples of taxonomies that were the result of similar theses and dissertations.
3. Taxonomies from software testing literature.

Taxonomies from software- testing literature.
Boris Beizer's "Bug Taxonomy" (Beizer 1990) [3]
Beizer provides his taxonomy in the book "Software Testing Techniques" which makes his taxonomy important in this context, as it is another taxonomy created for testing purposes. He divides his list into three types of bugs
1. Bugs in 'Design' phase
2. Bugs in 'Implementation/Coding' phase
3. Bugs in 'Maintenance phase'

He uses a 4-digit number to represent a bug and demarcate the levels.

Cem Kaner's appendix of "Common Software Errors" (Kaner *et al*.,)
Kaner *et al*., present their outline in the book "Testing Computer Software" 2nd edition. In his book, he suggests that the list be used for:
1. Evaluating test materials developed for you by someone else
2. Developing your own tests
3. To help replicate irreproducible bugs
4. Discover other bugs related to unexpected bugs you have just found.

Discussion: What constitutes a good taxonomy for testing purposes?
As seen earlier, the word "taxonomy" has been used in so many different contexts, that it is hard to pinpoint exactly one definition or define the requirements of what constitutes and qualifies to be called as a taxonomy.
Nonetheless, Lough (Lough 2001) [11] in his dissertation provides a list of properties. His combined list contains about 18 properties that a taxonomy should possess and it ranges from properties such as accepted, appropriateness, comprehensible, completeness etc to mutually exclusive, unambiguous and useful.
The e-commerce taxonomy is appropriate, comprehensible, specific, and most importantly useful for the purpose it was

created. It will possibly be accepted as a good beginning by the testing community and with possible customizations for individual needs, it may end up meeting its objectives.

The e-commerce taxonomy is far from being exhaustive or complete despite its mammoth span of categories and large number of failure modes. The e-commerce world and its systems are too diverse to have a single universal taxonomy that generalizes all potential failures across its different product lines, OS families and environments. The failure modes in this taxonomy only represent a sample of possible failures that the system can face. The tester is then further encouraged to conjure up other possible failures which are similar or different from the ones in the list and develop tests for them.

Another heavily debated feature of a taxonomy is that each category must be mutually exclusive to each other category. That is, the categories must not overlap. In the e-commerce world, it is common to find that multiple causes or symptoms for a bug may cause the bug to be overlapped across multiple risk categories. Hence the e-commerce taxonomy will probably never exist as a mutually exclusive taxonomy. During the early design days of the taxonomy, some people had pointed out that the inclusion of an "Others" category will render the taxonomy complete. Every bug will have a category in which to reside and if it does not, it will fit into the "others" category. Lough quotes Dr. Carl Landwehr, in suggesting the same idea but continues to question the usefulness and validity of the "other" category and calls its use debatable.

A taxonomy should always be expandable when a new category of risks is identified. An ever- evolving taxonomy tends to be more useful and current. As for the purpose of testing, it is very difficult to build a perfect taxonomy in the first attempt and available test time. But, a simple and broad categorization that is able to raise specific questions in the minds of the testers is sufficient for the purpose of using a taxonomy as an aid for test idea generation.

In summary a good taxonomy for testing purposes has enough detail for a motivated, intelligent newcomer to the area to be able to understand it, and is broad enough to raise at least a few issues new to someone with moderate experience in the area. A good taxonomy is a useful tool for informing a tester who is new to the area about the types of problems to be tested for.

Brainstorming Test Ideas

Challenges

- Lack of focus
- Lack of clarity
- Losing time
- Lack of structured framework
- Redundant ideas
- Unable to eliminate ideas that do not fit.
- Unable to locate a central idea
- Idea train stops
- Unable to inspire creativity
- Unable to identify the challenge
- Unable to define the issue
- Unable to induce lateral thinking
- Lack of paradigms
- Ideas: Large quantity and of low quality
- Lots of depth but no breadth in the ideas

Many of the challenges listed above could be mitigated if we had a organized structure to build our ideas upon.

Using Taxonomies to help brainstorm test ideas

During the writing of my thesis, I used to pose the question "what are the different ways do you think an e-commerce shopping cart can fail?" to testers, test consultants and test managers whom I used to met to discuss and get some ideas for my research. Sometime recently I posed the same question to a few of my colleagues and we did a mock test idea generation session with and without taxonomy. I have reproduced some of the test ideas we generated without using a taxonomy. This session lasted about 10 minutes and there were 3 testers. The testers have been in testing for an average of about 3 years and have not worked much on e-commerce testing. They had not seen the e- commerce taxonomy before.

Brainstorming session without a taxonomy

- Shopping cart does not load.(2)
- Unable to add item.
- Unable to remove item.
- Unable to modify order.
- Correct item not added.
- Shopping cart incompatible with browser and browser crashes.
- Hidden functionality, not able to find checkout button.
- Oops! Clicked the wrong button.
- Broken URLs.
- Missing URLs.
- Shopping cart fails to populate the images in the shopping catalogs.
- Able to hack the cart and change prices from client side.
- Customer credit card numbers compromised due to security glitch.
- Get "Page not found" error on clicking checkout button.

Comments and Observations

Positives

- Test ideas address some important areas of concern like security and functionality.
- Reflects user experience of commonly seen web issues.

Deltas (Areas needing an upward sign of improvement):

- Though the ideas address areas like functionality and security, other important areas like Performance issues, Accessibility, Scalability, Internationalizability (Qualitative issues) seem not to have caught the attention of the tester's idea train.

  Test ideas could have been broader and built around an idea framework to raise confidence levels that all areas of concern have been addressed.
- Description of some issues are very generic and clichéd, hence don't provide enough information to design tests to find them.
- Don't seem to address some media publicized or frequently seen issues such as holiday outages, payment processing glitches or glitches due to a failure at a third-party service provider.

- Redundant ideas, the numbers in the brackets indicate that the ideas were given more than once.

Observations

The generated list of test ideas has an ingrained structure for possible taxonomical risk categories. For example Incompatibility, Document security and Privacy (Ideas 12 and 13), Database Issues and Usability This suggests another way on how a simple taxonomy can built from a small collection of test ideas and can be used again with the same group to provide a structure to generate more focused and a stronger set of test cases.

Brainstorming session with the taxonomy

For this session I had given a simplified version of the e-commerce taxonomy to the testers (since we had decided to spend only 10 minutes, we decided that 45 categories were too exhaustive and testers will not be able to go through all of them within the allocated time)

Listed below are the test ideas generated with the taxonomy for each category. Time spent: 15 minutes to brainstorm.

Poor usability

- The user cannot add an item directly from the search result page.
- The user does not know at every single point in time how many items are in the cart and the total price.
- User has to go through too many pages to complete an order.
- Difficult to use the system: difficult to add, remove and update.
- Cannot to see the final value or estimate the checkout price.
- Hard to use the "Search" function and hard to locate the "Search" field.
- Unable to find "Help" menu
- Customer feedback forms unavailable.

Calculation/computation errors

- Removing/adding an item from the cart does not update the total.
- Negative number of items will discount from the total price.
- Shopping cart doesn't update/refresh price when adding new items.
- Discounts are not computed correctly.
- Postage fees or state taxes are not computed correctly.
- Recalculate function fails.

Internationalizability

- The registration fields do not accept extended/ international characters.
- If extended/international characters are entered into the registration, the database gets corrupted.
- Unable handle upgrades to a multi-lingual website.
- Unable to handle non-domestic orders and unable to integrate shipping costs for different countries.

Failure at ISP/Web Host

- The user successfully checks out but the notification e-mail never reaches him.
- Non-restorable data loss at hosting center.
- Back-up routines fail at hosting center and order data lost.
- Network Failures
- Link to inventory database goes down
- Link to user profile database goes down

Compliance

- Site does not follow HTML standard(W3C compliant)
- Non-compliant with possible credit card/ merchant account regulations.

Scalability

- The adding to cart, check out, and search processes take much longer during peak hours.
- Timeouts of requests during peak hours.
- Site cannot handle additional web/application/database servers

System security

- Test the strength of encryption.
- Test for vulnerability to buffer-overflow attacks.
- Test for vulnerability to SQL query attacks.

Client Privacy

- Check if there is a privacy policy
- Check for cookie expiration: check if anyone can access the content of the cart of a previous user (case of a shared computer)
- Test for existence of timeout routines that time-out the billing page when no activity is seen.
- Not able to opt out of customer profiling studies.

Web-server failure

- No custom error page in case of "Page not found error".
- Server fails under heavy load

Third-party software failure

- Failure of credit card verification system

Comments and Observations

- Under the given time, we see an increased number of good and more focused test ideas.
- Very structured and organized approach and the presentation tend to provide a sense of confidence and better coverage.
- Able to focus the tester's idea train to areas that have been identified to need more attention. Hence we now have a collection of test ideas which is more comprehensive and detailed.
- A few well known types of security attacks, Popular flaws were addressed when the testers were specifically prompted.
- It was agreed by the participants that the taxonomy helped them think more focused and the scope of the test idea generation session could be better understood. Also observed was that the taxonomy aided a smooth and organized facilitation of the entire session.
- We wasted less time and the whole exercise was more interesting than a traditional, unstructured test idea generation meet.

Ideas on facilitating a Brainstorming session using taxonomy.

Provided below is a mind-map that illustrates the process of rainstorming test ideas using a taxonomy.

Creating bug taxonomies for testers

How the e-commerce bug taxonomy was developed?

Brainstormed a first draft top-level list, and created a basic outline with about 10 categories, which formed the basis for further work. The categories partially passed the mutual-exclusivity test but were not broad enough. They did not completely cater to the needs of a tester because the emphasis was more on providing a framework to generate test ideas. I would assume here, that a tester will be able to generate better test ideas when the categories are specific and increase the number of test ideas when there are more categories. The next step was the generalization step of the reports of individual bugs I had collected. Some took shape as causal categories and others as symptomatic. But basically, just as.

Searched electronic bug databases (such as bugnet.com and cnet.com) for examples.

Bug databases (Bugtraq, Bugnet, etc), security advisories (CERT, MITRE, Securityfocus, etc), bug columns in magazines, newsgroups are good sources for actual bugs. Also Internet magazines such as eWeek, Cnet.com, Zdnet.com publish major e-commerce site outages, glitches and sometimes the cause for the outage as perceived by the industry experts or as released by the concerned e-commerce site.

Searched open source software for bug databases for specific products. These gave us examples and indications of the types of bugs possible.

Open source software sites are useful from the point of view that they make their bug databases public. It will be hard to find the complete list of the types of bugs that plague e-commerce databases/servers from commercial software sites such as Oracle or Sun. However, the bug database from a popular open source software site such as apache.org, which is available in the public domain, is useful.

Brainstormed additional types of problems.

ISO-9126 [7] categories were adapted with some modifications for the qualitative categories of the taxonomy. The final draft had about 60 categories. The process of changing, adding, removing and merging categories continued, however, until the present version of the taxonomy was created. The failure modes for the categories were inspired by ideas from troubleshooting guides, usegroups, talking to people who have tested these applications before, real-life bugs, vendor white papers, and risk forums.

Circulated the list and the outline for peer review.

The preliminary taxonomy and sample failure modes were circulated among peers, test managers, and users and any bug/issue/problem suggested by them went into the list.

I thank Karen Johnson, Bret Pettichord, Ross Collard and others for their valuable ideas.

Ideas on how you can create a simple and basic taxonomy specific to your application.

The pre-brainstormed taxonomy provides a whole lot of detail about possible failures to testers who will be working on similar applications. Generating risk list on the fly is usually inefficient but sometimes using a simple and basic outline to guide a brainstorming session was suggested to be useful by some people who have seen the e-commerce taxonomy.

Provided below are a few tips on how to possibly create a simple taxonomy in less time and use it to aid your brainstorming session.

You might have noticed that in an earlier section, where I made an observation about how an inherent taxonomy structure existed in the collection of ideas generated by the testers.

Hence one approach could be:

1. Identify a function that needs to be tested
2. Brainstorm a preliminary list of possible risks that could affect the system.
3. Now try to cluster the ideas into sub-categories and title them. Many testers, who participated in the study, commented that their thinking process worked in a hierarchical way, thinking ideas out for each possible failure category. If you can identify a pattern among the risks and create a category for each pattern, you will soon be having a preliminary taxonomy that you can further improve on.
4. You may want to expand on the preliminary categories to make more meaningful and specific categories. For example, if yourpreliminary risk list could be divided into four groups Security, Functionality, GUI, Performance. Then you can take each category and break them into more specific categories. For example, break Security into System Security, Client Security (Privacy) and Document security (Confidentiality issues).
5. Now check with the other testers and see if there are other categories they think should be included in the outline?
6. Use this outline of taxonomy for your next brainstorming session.

Other possible sources of information within your organization that could help you create a taxonomy are:

- Old bug reports from your bug database. Bug reports provide valuable insights into the types of commonly occurring bugs that have affected the system in the past. Referring the bug database to create/enhance your taxonomy also assures you that your testing will address past issues and that you will be checking to make sure that they don't creep into the system again.
- Talk to testers who have tested earlier versions of the same application (if one exists) and ask them questions like what types of issues did they commonly encounter? Did you find lots of issues with the backend? Was the system constantly under external attacks? Was memory leak the cause for most of the performance issues you reported? If yes, you have a category.

You may start with a small and simple taxonomy but set goals to improve on it. Taxonomies are ever evolving and they will

improve only with time. A taxonomy perfected iteratively may prove to be a valuable tool and may save time and also increase the quality of your test cases substantially.

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.

Failure Modes

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. The 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 theshopping 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 glitches” 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 incomplete installation that result in some security features being turned off.
- Software upgrades sometimes sets all options to ‘default’ automatically after the installation is complete. This in turn may overwrite 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

Scalability

Definition:

“The ease with which a system or component can be modified to fit the problem area” [IEEE 1990]

Failure Modes

Vertical Partitioning

“Vertical partitioning adds an additional layer to an application” (Skinner)

- Additional time delays due to the new processing layer placed ahead of a component
- Additional and unwanted maintenance overheads due to the new additional layer
- Queuing and De-Queuing faults may occur in implementations where a queue is added as a processing layer to scale up for the excessive load on some component
- Increase in design complexity due to addition of new layers
- Any failure in the additional layer may propagate through the base layers causing further failures.

Vertical Scaling

“Vertical scaling throws additional hardware at the application environment” (Skinner)

- Adding more hardware to an existing problematic piece may continue to give problems and add to cost of maintenance.
- An inability to improve performance once maximum scalability limit is achieved. The scalability graph begins to level up when the maximum limit of vertical scaling is reached.
- Performance problems may overrun the capacity available to deal with them.
- E-commerce architectures with poor layering may fail to scale up. Adding additional resources for each layer is easier and less expensive than adding resources to the whole structure.

Horizontal Partitioning

"Horizontal partitioning breaks a single logical component on a single server into several logical components on several servers." (Skinner)

- Costly overheads due to the division of resources that have dependencies
- Re-architecture of existing applications to create new logical components creates the risk of failure due to logical incompatibility, protocol mismatch, illogical data communication levels etcetera.
- Creation of new physical components may increase performance overheads and cost overheads.
- Possibility of administrative human error in a non-automated environment due to increased handling of monitoring and backup processes

Horizontal Scaling

"Horizontal scaling is the process of moving a single component into a "farm" of identical components" (Skinner)

- Failures in components that hold state information

Examples of Related Bugs and Issues

"A snippet that enunciates a horizontal scaling problem" (Skinner)

Problem

"Anyone who placed an order and subsequently went back to update the order could frequently not find that order. Also, when a customer changed an address on the Web site and later returned to place an order, the order was often associated with the old address.

Cause

"A technician explained that requests made by the Web site were randomly distributed across the six database servers. When a visitor returned to the site to view an order, there was only a one in six chance that the visitor's request would be routed to the correct database server.

The basic problem was that when a single database server was queried about an order, it needed (but did not have) the information held on the other database servers. Because of this, the database was not a good candidate for horizontal scaling." (Skinner)

Web Server Failures

"The Web server is the software that provides services to access a network, e.g., the Internet. A Web server hosts Web sites, supports HTTP and other protocols, and executes server-side programs."

A Web server failure is probably the most commonly assumed failure by users who experience a failure while surfing the Internet. Web servers have evolved from simple software that serves pages. Today they are sophisticated software that also hosts and supports numerous protocols, depending upon functionality, and may also run functional scripts. From the testing point of view, a Web server can display a multitude of symptoms depending upon its function, its failure and how it fails.

During the initial research I undertook a survey of available servers from different vendors. I reviewed their trouble shooting guides, white papers, technical support documents and open bug databases containing common bugs and issues. I would recommend that approach to any tester whose requirement is to write tests for a very specific brand of Web server such as IIS or Apache. Look up the product's site for known issues and troubleshooting guides for lists of common symptoms for potential failures. Some descriptions for the bugs were sometimes very succinct. The tester may have to know the server terminologies really well to understand them.

Provided below is a set of example failure modes for a general Web server.

Failure Modes

- Overflowing static buffer
- Remote users can execute UNIX shell commands in UNIX web servers
- Remote users can download CGI script executables
- Web server aborts during startup
- Memory leakage
- Header confusion - some headers appear twice, some never appear
- Incorrect handling of file names and types
- Incorrect file permission on the web server
- Incorrectly configured DNS
- Unable to update the DNS
- Missing secondary DNS lookups
- Some service daemons may be down, like ftp, and http may work but telnet does not work
- Exceed the maximum simultaneous connection limit
- Connections take significant amount of time to close- slow connections
- Failure of server side processes such as scripts, cgi or servlets
- Incorrect permissions on the scripts and server side executables
- Perl or shell scripts throw compilation errors
- Newer versions of interpreter run on older versions of scripts that may break the script
- Serves improper headers
- Outputs malform HTML
- Incorrect firewall or router configuration might cut access to the Web server
- Firewall resets after power outages to older incorrect configurations
- The HTTP daemon does not start at startup.
- Some web server processes have crashed or are not responding
- Server timeouts
- Resource intensive processes may cause server requests to timeout
- The machine on which the server is running may run out of resources such as CPU, memory, etcetera
- A non-Web server process may consume all system resources, choking the processes.
- Fast-growing log files consume all the disk space and a full disk may choke the Web server
- Inadequate Web server capacity
- Web servers buckle under peak load

Network Failures

A reliable network is key to the success of any e-commerce site. If the site is frequently offline for network maintenance, then it does not bode well for the maintenance staff and the business.

Network failures could be at both the ends of an e-commerce system-- the client side and the server side. The server side network issues are probably more in the control of the testers/troubleshooters than the client side one. A network issue could range from a total loss of connectivity to intermittent connectivity and even performance problems. The network risks that any IT infrastructure faces apply to an e-commerce infrastructure too. Listed below are few of the common risks/ failures that one can see in any network environment.

Failure Modes

- Node or link inoperative
- Node or link does not function
- Node or link continues to operate but incorrectly
- Incorrectly configured node
- Failures in underlying telecommunication switching systems
- Router failures
- Router table fails
- Router fails to scale to larger network implementations
- DoS attack on router
- Router fails to boot
- Device cannot establish IP communication with another device across a router, but can establish IP communication locally
- Network adapter fails at the server site
- Switch interface fails
- Transmit and receive cable pairs mismatch
- Load balancer fails
- Network hardware failures/ link failures
- Copper cables damaged/cut/ corrosion/magnetic interference
- Fiber cuts in fiber optic cables
- Cable breaks in Ethernet cables
- Network congestion
- Line card failures
- Slow restoration time
- Congestion resulting from re-routing
- Inefficient traffic engineering mechanisms
- Inefficient traffic engineering mechanisms
- Link failures and congestion
- Fiber cuts
- Line card failures
- Slow restoration time
- Congestion resulting from re-routing
- Inefficient traffic engineering mechanisms
- Issues related to network operation
- Issues due to platform upgrades
- Issues related to capacity expansion
- Configuration errors.
- Issues arising out of link expansion.
- Connectivity issues
  - ✓ Loss of connectivity
  - ✓ Intermittent connectivity
  - ✓ Timeout
- Degrading network performance.
- Duplicate addresses.
- Consistently high utilization rates

Hardware Failures

Hardware failures can occur anytime and anywhere. The only way to control the risk is by designing the hardware architecture with fewest possible single points of failure. In an e- commerce system, a hardware failure can broadly be classified as any of three types, depending upon where they occur: Server-side hardware failures, client-side hardware failures and third- party hardware failures.

Server-side failures are more within the realm and control of the testers and maintenance personnel, but the impact of hardware failures at a third-party service provider or at a client's site will seriously affect the completion of any transaction that is underway. Listed below are some common failures and risks impacting hardware systems. I have generalized them to cover the three types.

The three types of hardware failures were three separate categories in the original design of this taxonomy and I would encourage testers to consider them to brainstorm test ideas. Brainstorming where the failure occurs and how it impacts the system will be useful to prioritize on the test ideas.

Failure Modes

- A shopping cart may not function and pages may not be served if the hardware that runs any of the following servers fail:

Application server hardware

- Back-up server hardware
- Cache server hardware
- File server hardware
- Memory error may corrupt data on the hard disk and all shopping cart data, including the executable, web pages, scripts may be lost
- Non-recoverable disk failures and RAID (Redundant Array of Inexpensive Disks)
- failures
- Resource conflicts: Peripherals that try to use the same interrupt requests, DMA channels or I/O addresses may cause hardware resource conflicts; the e-commerce system may experience increased delay and even gradual failure
- High central processing unit (CPU) utilization by some processes and subsequent failure due to insufficient CPU
- RAM failures, paging faults, memory leaks in server hardware (refer to category on memory leaks for further failure modes)
- Inappropriate hardware configuration and problems due to insufficient resources. For example, creating the same hardware configuration used for a Web server for a back-up server. A back-up server may probably need more memory/disk space.
- Hardware time outs, which may lead to user session time outs during shopping sessions
- The shopping cart program tries to write to the wrong storage device

- The shopping cart program tries to read from the wrong storage device
- The shopping cart program may try to write into a wrong disk sector, over-write some other file on the disk and it may crash or time out
- Hardware fails due to over-heating
- Physical theft, physical damage, damage to server side hardware due to natural disaster, hardware damage due to water seepage/incorrect temperature control/ high humidity
- Hardware controllers may get corrupted
- System timing: Cache server hardware failures because access time is too fast to handle
- Power loss, power surge, intermittent power outages may cause short-circuits in mother board
- Modem failures due to power surges
- Ethernet card problems due to incompatibility with older machines
- Router failures
- Hardware-based load balancer fails
- Backup generators fails
- UPS fails to start upon power failure or starts only after the system reboots due to power loss
- Cables may get cut/damaged, Internet access devices may fail and the Internet access may get cut.

Navigation Failures

One of the most difficult problems in an e-commerce site design is navigation. If the navigation is bad, then:

- The user cannot find the content they are looking for quickly.
- They get lost and don't know where they are on the site.
- They lose context with the logic of the Website.

They get frustrated and you lose a valuable customer.

Listed below are some common failure modes for the navigation failure category.

Failure Modes

- Illogical placement and use of "next" or/and "back"/ "previous" buttons
- Bad design and selection of navigation structures or using non-linear navigation for linear segments of content
- Menu does not provide access to all segments of content.
- Content blind spot: some content not accessible by any of the navigation structures/paths.
- Information "buried" too deep; too many navigational "clicks" are required to get to the desired content segment.
- Navigational menu requires plug-ins that is incompatible with the user's browser.
- Inconsistent and unclear navigational aids
- Unable to dynamically change navigational structures to keep up with non-unique URLs generated by dynamic HTML pages
- Bad use of frames
- Presence of too many irrelevant commands on the page that might hamper the navigation flow by giving the user too many options to click on, which go down navigation paths that lead the user away from the task.
- Poor navigational design, which leads a user away from one page to another but when the user tries to come back to the older page, the state is changed/lost.
- Unclear or unable to navigate to the correct exit path.
- Navigational failures due to over-dependency on browser back button to control data, because novice users "do not know where the browser ends and the application begins." (Shubin and Meehan, 1996)
- Network delay and increased download time due to bad navigation design.

Third-party Software Failure

Many key services in an e-commerce site are provided by third-party services. The bill-payment processing is one such key service that is generally provided by a third-party. This implies that the third-party services should be failure-free in order for the e-commerce transactions to be processed to completion. But in reality there have been many instances where e-commerce sites have had to halt transactions because some subscribed third-party service has crashed. One indication of third-party failures could be the loss of one or more specialized functions from the website. For example in an online trading site, the stock values are pushed to the site by one service and the corresponding charts are drawn by a different third party service which specializes only in chart drawing. A missing chart could indicate that the third-party chart provider is down. Listed below are some failure modes, which exemplify this risk category.

Failure Modes

- Third-party bill processing or payment processing system fails due to unreliable batch processing, processor inadequacies or due to excessive load
- Third-party virtual server or dedicated server fails
- Password protection system is down due to a glitch in the third-party that provides the protection service

Changes to system configuration at the third-party location causes compatibility issues with e-commerce sites where it is deployed

- "Unable to process payment", probably the browser has javascript turned off or does not support it
- Data and communication hubs crash at the third party provider and the e-commerce site is affected
- The line to the third-party data center fails
- Third-party interfaces shut off some of the site's e-commerce features
- SSL web server certificate expires at the third-party bill processing unit; as a result the e- commerce site is unable to process payments

ISP and Web Hosting Problems

These are two services that can be called the backbone services to the entire concept of online business. One provides the connectivity to the outside world and the other hosts the site for the outside world to see. Hence any failure in these services leads to the site going totally offline to the users. Sometimes, because the same provider provides both these services, any failure can mean both loss of connectivity and the site being inaccessible.

Listed below are the common types of failures encountered by the e-commerce site due to problems in their ISP or web host.

Failure Modes
- Hosting platform crashes
- Insufficient data storage space in hosting server
- Decreased bandwidth and poor data transfer rate
- Poor site performance due to overburdened network connections
- Reduced simultaneous/concurrent transaction sessions due to limited network capacity
- Web host is unable to accommodate high traffic volume
- One of the network services is disrupted and the web host is unable to redirect requests to alternate services, as the network design is non-redundant
- Failures along the Internet backbone
- DNS entries are not pointing to the right IP address
- All communication ports closed by mistake and system needs reboot
- Growing log files of one site consumes disk space and chokes other sites in a shared server hosting arrangement
- Frequent restoration of files from backups with no notification
- Web host does not work with secure connections but works well with regular http connections
- Significant planned downtime with no advance notification
- Hosting server dies with no errors
- Kernel-level bugs, kernel panic in Linux based host systems

Bad RAM
- Badly configured configuration tables, IP tables, and random services blocked off
- Server ran out off swap space
- Blacklisted IPs

IP conflicts
- Nameserver fails

Issues due to Non- Compliance
Definition:
"Attributes of software that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions." (ISO 9126: 1991) [7]

Failure Modes
- Non-compliance with usability guidelines
- Non-compliance with federal accessibility guidelines
- Omission of basic information under Distance Selling Regulations
- Lack of compliance with Data Protection Act
- Standard terms of business omitted
- Inadequate site design leading to legal notices not being part of the contract
- Committing to supply goods when item is not in stock
- Misuse of others' intellectual property rights, like using competitor's name in meta tags
- Exposure to libel where the public can add text to your site e.g. on bulletin boards
- Non-compliance with restrictions on the type of conducting a particular type of online business such as online gambling, sale of restricted drugs without prescription, pornography, tobacco etcetera
- Non-compliance with security guidelines that lay rules with respect to encryption keys
- Non-compliance with licensing regulations that are needed to sell certain types of products
- Non-compliance with domain name registration rules and country level domain names registration rules
- Scroll bars fail to work in "Clickwrap" agreements, and user is unable to scroll and read all sections of the agreement inside the wrap text.
- Non-compliance towards digital signature authentication guidelines
- Non-compliance with privacy laws
- Failure to post a privacy disclosure statement
  - ✓ Europe: Data Privacy Directive
  - ✓ Malaysia: Personal Data Protection Bill
  - ✓ Japan: comply and seek "Privacy Mark"
  - ✓ Singapore: E-commerce code for the protection of personal information and communications of consumer of E-commerce
- Non-compliance with advertising standards, untruthful, false and misleading ads
  - ✓ Japan: Prevention of unreasonable premiums and misleading representations concerning products and services
  - ✓ Philippines: The Consumer Act
  - ✓ Thailand: Consumer Protection Act 1979
- Violation of consumer protection laws that include anti-fraud, advertising, usury, Installment contracts and rebate standards
  - ✓ Japan: Direct Sales Law, 1976
  - ✓ Japan: Law on Consumer Contracts, 2001
  - ✓ Korea: Basic Law on e-commerce, July 1999
  - ✓ Taiwan: Consumer Protection Law (CPL), 1994
  - ✓ New Zealand: eMarketing Standards Authority
- Violation of content control regulations imposed by many countries
  - ✓ Germany: Article 131 of German Penal code
  - ✓ Germany: Communications Services Act
  - ✓ France: Article R.645-2 of the French Criminal Code
  - ✓ China: "Tentative Provisions", Feb 1996
  - ✓ China: Computer Information System Internet Security Administration – Jan
  - ✓ 2000
  - ✓ Hong Kong: Control of Obscene and Indecent Articles Ordinance (COIAO)
  - ✓ Singapore: Internet Code of Practice
  - ✓ Australia: Broadcasting Services Amendment (Online Services) Act Jan-2000
  - ✓ New Zealand: Voluntary code of practice for Internet service providers
- Non-compliance with electronic interconnection standards for e-commerce/ IPC-2500 series

Understandability
Definition
"Attributes of software that bear on the users' effort for recognizing the logical concept and its applicability" (ISO 9126) [7]

Failure Modes

- 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 as this leads to the shopper getting confused/uninterested, causes shopping cart abandonment.
- Check if thumbnail photos of the items can be added to the shopping carts in addition to a text description. This reassures the customer that the right item has been added to the shopping cart.
- 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 if the UI provides functionality for discounts and coupons. Provide separate field in the UI to display discounts due to coupons as 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; avoid this common problem

Learnability

Definition

"Attributes of software that bear on the users' effort for learning its application (for example, operation control, input, output)" (ISO 9126) [7]

Failure Modes

- Try not using pop-up window based shopping carts because if the user clicks elsewhere in the main window, the pop-up is sent "behind" the main window.
- Provide "remove item" or "add item" buttons instead of asking the user to change "item quantity number" as it is easier and more error free.
- 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?
- Check if the "Continue Shopping" and "Proceed to Checkout" buttons are visible.
- Do not limit the features of the shopping cart--keep it flexible.
- Cart is too hard to use. Solution: reduce functional complexity.
- 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.

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.

Operability

Definition

"Attributes of software that bear on the users' effort for operation and operation control" (ISO 9126) [7]

Failure Modes

- 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.
- Look for items that have not been linked back to the "item"/catalog page.
- 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, test if you are able to return back to the shopping cart from the detailed page and check if the state of the shopping cart is maintained.
- 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 on the right as users perceive it analogous to "back" and "going forward" respectively.
- 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 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! This 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, and erratic placement of multiple selection checkboxes
- Check if shipping can be calculated before checkout. Shoppers prefer getting an idea of the total cost of the item.

References

1. Ande, James P. Anderson. Computer Security Threat Monitoring and Surveillance. Technical Report Contract 79F296400, James P. Anderson Co., Box 42 Fort Washington, PA 19034. 1980; (215):646-4706.
2. Aslam, Taimur Aslam. A Taxonomy of Security Faults in the UNIX Operating System. Master's thesis, Purdue University. 1995.

3. Beizer, Boris Beizer. Software Testing Techniques. Van Nostrand Reinhold, 115 Fifth Avenue, New York, New York 10003, second edition. 1990.
4. Howard, John Howard D. An Analysis of Security Incidents on the Internet. Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 USA. 1997.
5. Jayaram, Jayaram ND, Morse PLR. Network Security — A Taxonomic View. In European Conference on Security and Detection. School of Computer Science, University of Westmister, UK, IEE, 28–30 April 1997. Conference Publication No. 1997, 437.
6. IEEE. Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY. 1990.
7. ISO. ISO/IEC 9126-Information Technology, Software Product Evaluation, Quality, Characteristics and Guidelines for their Use. 9126.
8. Kaner *et al*. Kaner, Falk, Nguyen. Testing Computer Software John Wiley 2nd edition. 1999.
9. Krusul. Ivan Victor Krsul. Software Vulnerability Analysis. Ph.D. dissertation, Purdue University. 1998. http://www.krsul.org.
10. Lindqvist. Ulf Lindqvist and Erland Jonsson. How to Systematically Classify Computer Security Intrusions. In IEEE Security and Privacy, pages 154–163, Department of Computer Engineering, Chalmers University of Technology, SE-412 96 G¨otegorg, Sweden. 1997.
11. Lough, Lough DL. A Taxonomy of Computer Attacks with Applications to Wireless Networks. Virginia Polytechnic Institute and State University. 2001.
12. Meehan, Shubin. Meehan, Shubin. Navigation in Web Applications. 1996.
13. Neumann, Parker, Peter Neumann G, Donn Parker B. A Summary of Computer Misuse Techniques. In 12th National Computer Security Conference. 1989, pages 396-407.
14. Richardson. Thomas Winfred Richardson. The Development of a Database Taxonomy of Vulnerabilities to Support the Study of Denial of Service Attacks. PhD thesis, Iowa State University. 2001.