1. Given an Amazon Shopping Cart with 5 items, provide 10 test cases for each of the following test areas:
   1. Functional (remove an item, proceed to checkout, etc…)
   2. UI (buttons, text boxes, links, etc…)
   3. Pick your favorite that is not #1 or #2

1. What would you include in a test plan for the above Amazon shopping cart?
   * 1. Give a brief summary with a targeted purpose for each item you would include.

1. Answer at least 2 of the following questions:
   * 1. What test exclusions would you call out in your test plan?
   1. What is the responsibility of the developers in your test plan?
   2. What Security testing would you do in the shopping cart?
2. Given an Amazon Shopping Cart with 5 items, provide 10 test cases for each of the following test areas:
   1. Functional (remove an item, proceed to checkout, etc…)
3. **Start by removing all 5 items. Confirm removal is successful.**
4. **Add a Single Product to Cart**
5. Select a product.
6. Select Add to Cart.
   1. Verify that the correct product display page - PDP - is showing all necessary links and information.
   2. Verify product successfully added to cart and in the Cart icon.  Only 1 item is displaying.
7. **Added multiple Products to Cart**

1. Select a product.

2. Select Add to Cart – add the First Product.

3. Select Add to Cart – add Second Product.

1. Verify that the correct product display page - PDP - is showing all necessary links and information.

2. Verify product successfully added to cart.

3. Verify two products are showing in the cart and in the cart icon, 2 should be displaying.

1. **Validate Cart after editing Quantity**

1. Select a product.

2. Select Add to Cart – add the First Product..

3. Select Add to Cart – add Second Product.

4. Editing the quantity of items in the cart with valid and invalid quantities (like spaces, negative values, characters, 0, etc).

1. Verify that the correct PDP page is displaying all the necessary

links and information.

2. Verify product is successfully added to cart.

3. Verify the items quantity is successfully edited. If user gives a -ve value displayed a warning message to user.

1. **Cart – Validate Backend sync**

1. Select a product with an inventory count that is less than 5 (to be validated at backend)

2. Select Add to Cart icon to add the Product.

3. Edit the quantity of the product in cart with a 5 and attempt to checkout.

1. Verify that the correct product display page - PDP - is showing all necessary links and information.

2. Verify product successfully added to cart and in the Cart icon.

3. Verify if items quantity count is greater than the actual item’s inventory count then checkout should prevent the user from proceeding and the proper error should be presented to the user.

1. **Remove a single product from Cart**

1. Select a product.

2. Select Add to Cart.

3. Select the Cart.

4. Select the Cross symbol to remove the product from the cart.

1. Verify that the correct product display page - PDP - is showing all necessary links and information.

2. Verify product successfully added to cart and in the Cart icon.  Only 1 item is displaying.

3. Verify the product was added and it’s displaying correctly in the cart

4. Verify the Product is removed from cart and Cart icon is displaying 0 items.

1. **Remove multiple products from Cart**

1. Select a product.

2. Select Add to Cart – add the First Product.

3. Select Add to Cart – add Second Product.

4. Select the Cross symbol to remove any and all of the product from the cart.

1. Verify that the correct product display page - PDP - is showing all necessary links and information.

2. Verify product successfully added to cart and in the Cart icon.  Only 1 item is displaying.

3. Verify both products are showing in the cart and that the Cart Icon is displaying 2 items only.

4. Verify the product has been removed from the cart and Cart icon is displaying 1 items.

1. **BuyNow**

1. Select a product.

2. Select the Buy Now option.

1. Verify that the correct product display page - PDP - is showing all necessary links and information.

2. Verify Product purchase page is showing all available payment options should to the user.

1. **Guest Checkout**

1. Launch the test app - Don’t sign in.

2. Select a product to buy.

3. Now sign-in and Checkout.

1. Validate the test app is successfully launched and a msg of Hi Guest is clearly displaying at for the user.

2. Verify the correct PDP page includes all the payment options should showing to the user.  The app should ask the user to login.

3. Verify user can successful checkout.

1. **Validate CheckOut for a Registered user**

1. Start the test app.

2. login.

2. Select a product to buy.

3. Checkout.

1. Verify test app is successfully launching and that Hi, Username is dispalying.

2. Verify that the correct PDP page with all the payment options are shown to the user.

3. Validate user is able to successful checkout.

1. **Gift Coupon Validation**

1. Select a product which allows a Gift Coupon.

2. Confirm the Coupon is valid while checking out.

1. Verify Gift Coupon is valid and that it’s code is tagged to the product.

2. Confirm that the amount of the Gift Coupon is successfully deducted at checkout.

* 1. UI (buttons, text boxes, links, etc…)

1. Make sure the **buttons** are all there
2. Make sure all the **text** **boxes** are there
3. Make sure all the **colors** are correctly displayed on various devices.
4. Make sure no **spelling** mistakes
5. Make sure all **links** are valid
6. Make sure **text boxes** are able to lower and upper case numbers
7. Make sure all **images** are shown correctly
8. Make sure all **tabs** are working
9. Make sure **scroll** on page works
10. Make sure **zoom** and zoom out is working
11. Make sure **color**-blind people are able to navigate site
12. Make sure **page titles** are correctly displayed.
13. Emails on page are working.

* 1. Pick your favorite that is not #1 or #2

(This isn’t really clear. Does this mean not functional nor UI? If so below is my answer)

**Performance (**how well a software can handle user traffic**)**

I would test the performance of the website. The website needs to load quickly to retain customers otherwise customers will get frustrated and move to other sites. There are many tools to use to test the performance of a website.

Breaking it down to:

* + **Load testing:** how quickly does it load when one then 15 then 1000 then 1000 ..etc users visits the website
  + **Stress testing**: to challenge the limits of the application e.g. if you get a billion users! (which is outside the sites limits) What happens! Crash?!?

Some of those performance tests are:

1. Have 100 users press **add** item at the same time.
2. Have a 100 users press **change** amount at the same time
3. Have a 100 users press **buyNow** at the same time
4. Have a 100 users press add item and leave it for a week without buying and come back to it.
5. Have a 100 users press **purchase** at the same time. (buyNow is different from purchase)
6. Have a 1000 users **enter their credit card information** and press next at the same time
7. Have a 10000 users **cancel** their purchase at the same time.
8. Have a 100 users enter **the same coupon** at the same time
9. Have a 1000 users **proceed to checkout** at the same time.
10. Have a 1000 users **order the same product** (which you only have one item of) at the same time.
11. **Put this test in a loop**:
12. Add an item to the cart
13. Then Browse for another item
14. Add s second item to the cart
15. Go to checkout
16. Complete the checkout process

Then watch for performance bottlenecks. This will help tune and remove bottlenecks.

1. What would you include in a test plan for the above Amazon shopping cart?
   1. Give a brief summary with a targeted purpose for each item you would include.

Basics of Testing an e-Commerce Website:

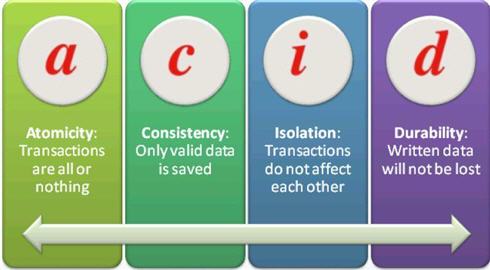
<https://dzone.com/articles/8-cases-to-skyrocket-your-e-commerce-website-testi>

e-commerce sites and applications or mobile applications undergo:

* A/B testing
* Usability Testing (User Acceptance Testing - UAT) -
* Database Testing
* [UI Testing](https://www.testing-whiz.com/blog/how-to-overcome-ui-automation-testing-challenges?utm_source=Quora&utm_medium=QnA&utm_campaign=Rachana&utm_content=Content)
* [API Testing](https://dzone.com/articles/10-effective-ways-for-successful-api-testing?utm_source=Quora&utm_medium=QnA&utm_campaign=Rachana&utm_content=Content)
* Mobile Application Testing
* Security Testing
* Performance Testing
* Load Testing
* [Functional Testing](https://www.testing-whiz.com/blog/6-essential-factors-for-successful-functional-test-automation?utm_source=Quora&utm_medium=QnA&utm_campaign=Rachana&utm_content=Content)

**In Details**

* A/B testing (**split testing** or **bucket testing**) - is a method of comparing two versions of a webpage or app against each other to determine which one performs better.
* Usability Testing (User Acceptance Testing - UAT) - is a method by which users of a product are asked to perform certain tasks in an effort to **measure** the product's **ease-of-use**, **task time**, and the **user's perception** of the experience.
* Database Testing –
  1. Data Mapping:
     1. are the UI fields mapped to the data fields?
     2. Correspondence between UI actions and [CRUD (Create, Retrieve, Update and Delete)](https://www.softwaretestinghelp.com/database-testing-process/)
  2. ACID properties validation:
     1. **Atomicity**
     2. **Consistency**
     3. **Isolation**
     4. **Durability**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2013/08/DB-Testing.jpg)

* 1. Data integrity
  2. Business rule conformity:
* [UI Testing](https://www.testing-whiz.com/blog/how-to-overcome-ui-automation-testing-challenges?utm_source=Quora&utm_medium=QnA&utm_campaign=Rachana&utm_content=Content) GUI testing
* [API Testing](https://dzone.com/articles/10-effective-ways-for-successful-api-testing?utm_source=Quora&utm_medium=QnA&utm_campaign=Rachana&utm_content=Content) API testing is a type of software testing that involves testing application programming interfaces directly and as part of integration testing to determine if they meet expectations for **functionality**, **reliability**, **performance**, and **security**. Since APIs lack a
* Mobile Application Testing
* Security Testing
* Performance Testing
* Load Testing (such as load runner, Visual Studio Enterprise edition,

Several types of load testing are employed

* Static testing is when a designated constant load is applied for a specified time.
* Dynamic testing is when a variable or moving load is applied.
* Cyclical testing consists of repeated loading and unloading for specified cycles, durations and conditions.
* [Functional Testing](https://www.testing-whiz.com/blog/6-essential-factors-for-successful-functional-test-automation?utm_source=Quora&utm_medium=QnA&utm_campaign=Rachana&utm_content=Content)

Before listing the test plan it’s important to understand the big picture, i.e the funnel of an e-Commerce website:

The traditional e-commerce funnel has 5 stages:

1. Acquisition: This is the first step. It’s were the user visits an e-commerce website
2. Engagement: The user is browsing the product(s) or service(s) catalogue, and they start evaluating existing options. Thereafter the user starts to consider buying by adding items to the shopping cart
3. Activation: The user is now set on purchasing so the user checkouts the desired items and makes the payment.
4. Retention: Is when a user is back at the website the user visits the e-commerce website or app and shops again.
5. Referral: The customer/user is satisfied with that the post purchasing service was good enough to recommend. This means the user will return to the website and will advocate for the website.

Make sure Clicks, registrations, payments, subscriptions all works.

**Requirements**

User should be able to add, remove, and change quantities of the items in their cart.

User should be able to get back to item pages from the cart by clicking on individual cart item lines.

User should be able to put in their address information to get shipping information estimates.

User should be able to add coupons, promocodes which and website should confirm if codes are expired or not.

Pay attention to cases such as logging in (if user has items in their cart as an anonymous user, but also have cart items from a previous authenticated session), or adding another item of the same SKU as others in their cart.

**Shopping Cart Test Plan**

**Pages of the Product category**

**If user clicks on any product in the basket should be taken to product. In the product:**

* Filters such as product or items in basket filters, colors, sizes, types of product, etc.
* Sorting products or items in basket by name, price, size, etc.
* Add/Remove form basket to a shortlist or wish list facility.
* Add/Remove from basket.
* Tools tips work.
* Drop down menus work
* Select/deselect
* Scrolling? Auto scroll?
* Hover function?
* Rendering on different browsers and different resolutions.

**Shopping basket**

* Add product(s) to shopping cart.
  + - * Item should go added
    - Empty Cart
      * When I click add from a product page
      * Then the cart should show a new line item
* Product Info images ..etc displayed correctly.
* Links between product in cart and product page works.
  + - * If cart has an item
      * Clicking the item link should take user to item page
* Change quantities of product in shopping cart.
  + - * Then quantity should update with new quantity
      * New price should show
      * Decrease quantity below 0
      * Use should be warned
      * When user adds another item of the same sku
      * User should have 1 cart line item with quantity increased
      * User should not have 2 copies of the same item in the cart
* Remove product from shopping cart.
  + - * Item should go deleted
      * When user clicks on an item link for a deleted item (sad path)
      * User should get a 404 page or proper error msg
* Check error messages if no items are in shopping cart.
* Check discounts, taxes and delivery costs are correct etc
* Min order works with free delivery
* Subtotal adds accurately.
* Valid discount code added
* Invalid discount code added
  + - * Given user can view the cart
      * When user input a coupon containing a SQL injection
      * Then the database should remain unpwned
* After purchase correct links, return policy …etc is displayed.
* Click Checkout or Pay Now to process to Checkout process.
  + - Pay attention to cases such as logging in (if user has items in their cart as an anonymous user, but also have cart items from a previous authenticated session), or adding another item of the same SKU as others in their cart.
    - Pay attention to cases such as logging in (if user has items in their cart as an anonymous user, but also have cart items from a previous authenticated session), or adding another item of the same SKU as others in their cart.

**Social Media**

Given an item listed in Social Media are you able to click on it and be taken to the site/app and the cart lists the item(s).

**Mobile Device Compatibility**

Does it work on mobile devices (compatibility)?

* Various phones
* Various browsers on phones
* Apps on different phones

**Browser Compatibility**

Various browsers testing is essential:

* Internet Explorer, MS Edge
* Mozilla Firefox (latest version)
* Google Chrome (latest version)
* Safari

**Checkout and Payment Systems**

Check the payment system of the products selected.

Process:

* Guest checkout works.
* Existing customers able to login.
* Existing customers can login through social networks.
* Users can register a new account.
* Delivery address details can be entered.
* Alternative billing address are available and can be entered.
* User able to select payment method.
* Final payment amount is correct, after the taxes coupons …etc
* Test making changes to the products being ordered, changing delivery options, etc. and make sure that this final amount updates correctly.

**Payment System**

Test payment using each payment method that are being offered such as debit cards, credit cards, Paypal, Google Checkout, etc.

An example list of payment types is as follows:

* Place Paypal payment
* Place Visa payment
* Place Visa Debit payment
* Place Visa Electron payment
* Place Mastercard payment
* Place Amex payment
* Place false payment
* Test cancelling order
* If a user is signed in for a long time, ensure there is a session timeout. All websites have different thresholds.
* Message/email confirmation with the order number that is generated after the purchase is made.
  + - The Credit Card
  1. check the credit card types allowed
  2. check the expiration dates of the card whether adding or
  3. updating the card info
  4. check the credit card billing address whether the address is
  5. updated or the card is updated

**Shipping:**

* Shipping address information

If user did NOT input address information

When user clicks to check out

User should be asked for address information

If user had no input address information

When user views the cart

Then there should not be a shipping estimate

If a user didn’t input address information

When user input address information

Then user should see a shipping estimate

**Login Handler:**

Given user not logged in

When user add an item to cart

Then user should be able to view the item in my cart

Given a user is not logged in

When user adds an item to cart

And user log in

And user add another item to the cart

Then user should see both items in the cart

Given user is logged in

When user add an item to the cart

And user log out

Then user should not see the item in the cart

**Check Analytics**

Make sure check analytics are installed for the shopping cart.

1. Answer at least 2 of the following questions:
2. What test exclusions would you call out in your test plan?
3. What is the responsibility of the developers in your test plan?
4. What Security testing would you do in the shopping cart?

a. What test exclusions would you call out in your test plan?

1. Exclude old OS systems
2. Exclude unpopular or unsupported browsers
3. Exclude unpopular or unsupported phones apps
4. Exclude unpopular or unsupported phones browsers
5. Feature that are not included in this version of the software
6. Feature that are low-risk
7. Features that are not impacted by current changes
8. products that are no longer supported
9. Products that have already been tested and not selling.

b. What Security testing would you do in the shopping cart?

[Security Testing Test Scenarios URL](https://www.softwaretestinghelp.com/sample-test-cases-testing-web-desktop-applications/)

1. Check for SQL injection attacks.

2. Secure pages should use the HTTPS protocol.

3. Page crash should not reveal application or server info. Error page should be displayed for this.

4. Escape special characters in the input.

5. Error messages should not reveal any sensitive information.

6. All credentials should be transferred over an encrypted channel.

7. Test password security and password policy enforcement.

8. Check application logout functionality.

9. Check for Brute Force Attacks.

10. Cookie information should be stored in encrypted format only.

11. Check session cookie duration and session termination after timeout or logout.

11. Session tokens should be transmitted over a secured channel.

13. The password should not be stored in cookies.

14. Test for Denial of Service attacks.

15. Test for memory leakage.

16. Test unauthorized application access by manipulating variable values in the browser address bar.

17. Test file extension handing so that exe files are not uploaded and executed on the server.

18. Sensitive fields like passwords and credit card information should not have to autocomplete enabled.

19. File upload functionality should use file type restrictions and also anti-virus for scanning uploaded files.

20. Check if directory listing is prohibited.

21. Password and other sensitive fields should be masked while typing.

22. Check if forgot password functionality is secured with features like temporary password expiry after specified hours and security question is asked before changing or requesting a new password.

23. Verify CAPTCHA functionality.

24. Check if important events are logged in log files.

25. Check if access privileges are implemented correctly.

Penetration testing test cases – I’ve listed around 41 test cases for penetration testing on this page.

1. ~~Use stolen cc numbers~~
2. ~~Use false names with cc no~~
3. ~~Use false cc nos~~
4. ~~Use wrong addresses with cc~~
5. ~~Use wrong zip code with cc no~~
6. ~~Make part of the address to be wrong~~
7. ~~Make part of the name to be wrong~~
8. ~~Use cc nos less than what a American express would have~~
9. ~~Select American express for a cc and enter numbers for a visa cc~~
10. ~~Make part of the zip code to be wrong~~
11. SQL Injections,
12. Ethical hacks on Login,
13. Register,
14. Payment gateway, and other various pages.
15. Cookies Security settings Testing
16. Registration:
    * 1. Account:
         1. Unique user ID
         2. Each user has a unique account
         3. Passwords
            1. Are they stored encrypted?
            2. Are they encrypted in user logs?
17. SSL
    * 1. Is the registration process secure?
      2. Can user break out from registration?
      3. SSL adds numerous features to that stream, between two computers or processes that are exchanging data. The TCP/IP protocol simply sends anonymous error-free stream. SSL encrypts that data exchange.
         1. Authentication and non repudiation of the server, using digital signatures
         2. Authentication and non repudiation of the client, using digital signatures
         3. Data confidentiality through the use of encryption
         4. Data integrity through the use of message authentication codes
18. Dd
19. D
20. Cookies: What to test
    1. Test your site’s cookies according to the content of the cookie.
    2. Test with cookies and without.
    3. Test the expire date, domain, path and security setting.
    4. Test with Internet Explorer and other browsers
21. Customers
    1. What information is maintained about each customer?
    2. Is your site customized?
    3. What information is stored encrypted?
    4. What are the rules for sending emails?
22. Session
    1. Session timeout
    2. Unique session IDs
    3. Session caching
    4. User logs
23. Servers
    1. Server re-directs
    2. SSL
    3. Configuration files
    4. User logs
24. d