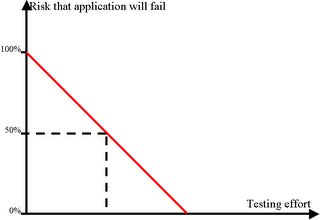
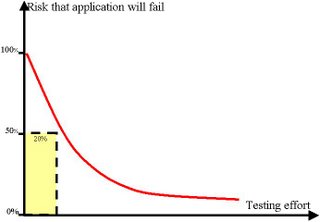
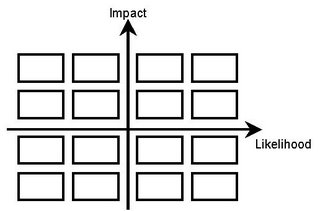
**Risk-Driven Testing / Risk Based Testing**

Why?  
Whenever there's too much to do and not enough time to do it, we have to prioritize so that at least the most important things get done. In testing, there's never enough time or resources. In addition, the consequences of skipping something important are severe. Classify features or application unit with severity ranking, defining severity by measuring the negative impact of an eventual issues that could have on the work of that product.

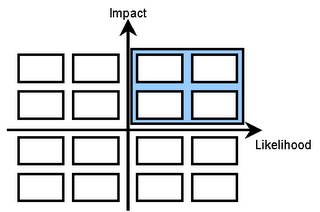
***Example of severity:*** *High - meaning the customer cannot work with the software and wastes time and money, medium - meaning the user has to use some alternative workaround to achieve the aimed goal and low - meaning the customer can STILL work with the software.*

This is classic theoretical relation between Risk and Testing effort:  
  
[](http://photos1.blogger.com/blogger2/3528/551481411325314/1600/01.1.jpg)It is known that usually 20% of the total features of an application will allow user to do 80% of his work (Pareto Principle in software products). So the ideal relation between Testing effort and risk will look something this:  
[](http://photos1.blogger.com/blogger2/3528/551481411325314/1600/02.jpg)  
where most of the problem has been discovered within first 20% of testing effort.  
  
To make it real, we have to make sure we test what's most prior first and only after that, if we have time, we can attend to things that are less important. This is why prioritization has received a lot of attention.  
This type of testing is called "Risk Driven Testing", where Risk has very specific meaning.

How?  
Take the pieces of your system, whatever you use - modules, functions, section of the requirements even test cases and scenarios - and rate each piece on two variables, Impact and Likelihood.  
Impact - is what would happen if this piece somehow malfunctioned. Would it destroy the customer database? Or would it just mean that the column headings in a report didn't quite line up?  
Likelihood - is an estimate of how probable it is that this piece would fail.  
Together, Impact and Likelihood determine Risk for the piece.  
Placing all pieces of product on virtual axes of likelihood and impact, you will obtain something like this:  
[](http://photos1.blogger.com/blogger2/3528/551481411325314/1600/03.jpg)Having this diagram we will separate our testing effort in 4 big steps.

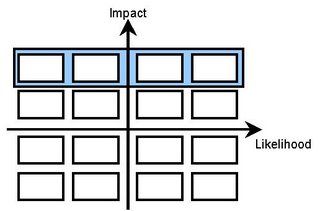
***Example:*** *You received from the development team a new application; this application was never been tested before. And you have been told that you have to test it, they also say they don't know how much time you have to test it. You can have a month or you can have a day, every day may be last day for testing.  
This is a simple application that enables you to save some objects and their proprieties. Also user can edit an existing record and can delete an existing record. Application enables you to delete and edit multiple records. All changes that have been made to the objects including deleting and adding a new object are logged. Objects are displayed to the user; user can select to clipboard their proprieties and can sort the object list and search objects by different criteria. Multiple selection is done by dragging, imagine cells in excel.  
Think of the application as a desktop application that connects to the server that has the database.  
Please make abstraction of the other details that I didn't mention, or if you think you can't continue without some details, make your own assumptions.*

First step

The pieces in the upper right hand quadrant, which have both high Impact and high Likelihood, need the most of our attention. That's why these pieces will be tested first.  
[](http://photos1.blogger.com/blogger2/3528/551481411325314/1600/04.jpg)

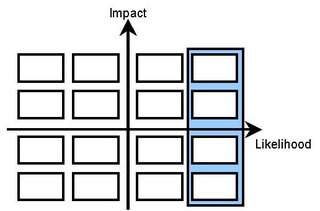
***Example:***- stores objects and their proprieties;  
- delete objects and their proprieties, one by one;  
- displays existing records;

**Second step**

Each piece of the product which is considered to have a higher Impact require our attention, even if the Likelihood value is low.  
[](http://photos1.blogger.com/blogger2/3528/551481411325314/1600/05.jpg)

***Example:***- edit existing records, one by one;  
- multiple delete of the objects and their proprieties;

Third step

In this step we attend to the things that are most likely to happen, regardless its impact.  
 [[](http://photos1.blogger.com/blogger2/3528/551481411325314/1600/06.jpg)](http://photos1.blogger.com/blogger2/3528/551481411325314/1600/06.jpg)

***Example:***- changes logging;  
- search objects;

Fourth step

In this final step, in case you have enough time for it, you will be testing the rest of the application. Those modules/pieces that have a smaller impact on the product overall and are very unlikely to be encountered by the user.  
Here goes those 80% of the test effort that I've mentioned earlier.

***Example:***- sort objects;  
- multiple edit;  
- clipboard selection;  
...