

<p>What exactly does a software tester do?</p> <p>Apart from exposing faults (“bugs”) in a software product confirming that the program meets the program specification, as a test engineer you need to create test cases, procedures, scripts and generate data. You execute test procedures and scripts, analyze standards and evaluate results of system/integration/regression testing. You also...</p> <ul style="list-style-type: none">Speed up development process by identifying bugs at an early stage (e.g. specifications stage)Reduce the organization's risk of legal liabilityMaximize the value of the softwareAssure successful launch of the product, save money, time and reputation of the company by discovering bugs and design flaws at an early stage before failures occur in production, or in the fieldPromote continual improvement <p>What makes a good tester?</p> <p>As software engineering is now being considered as a technical engineering profession, it is important that the software test engineer’s posses certain traits with a relentless attitude to make them stand out. Here are a few.</p> <ul style="list-style-type: none">Know the technology. Knowledge of the technology in which the application is developed is an added advantage to any tester. It helps design better and powerful test cases basing on the weakness or flaws of the technology. Good testers know what it supports and what it doesn’t, so concentrating on these lines will help them break the application quickly.Perfectionist and a realist. Being a perfectionist will help testers spot the problem and being a realist helps know at the end of the day which problems are really important problems. You will know which ones require a fix and which ones don’t.Tactful, diplomatic and persuasive. Good software testers are tactful and know how to break the news to the developers. They are diplomatic while convincing the developers of the bugs and persuade them when necessary and have their bug(s) fixed. It is important to be critical of the issue and not let the person who developed the application be taken aback of the findings.	<ul style="list-style-type: none">An explorer. A bit of creativity and an attitude to take risk helps the testers venture into unknown situations and find bugs that otherwise will be looked over.Troubleshoot. Troubleshooting and figuring out why something doesn’t work helps testers be confident and clear in communicating the defects to the developers.Posses people skills and tenacity. Testers can face a lot of resistance from programmers. Being socially smart and diplomatic doesn't mean being indecisive. The best testers are both -socially adept and tenacious where it matters.Organized. Best testers very well realize that they too can make mistakes and don’t take chances. They are very well organized and have checklists, use files, facts and figures to support their findings that can be used as an evidence and doublecheck their findings.Objective and accurate. They are very objective and know what they report and so convey impartial and meaningful information that keeps politics and emotions out of message. Reporting inaccurate information is losing a little credibility. Good testers make sure their findings are accurate and reproducible.Defects are valuable. Good testers learn from them. Each defect is an opportunity to learn and improve. A defect found early substantially costs less when compared to the one found at a later stage. Defects can cause serious problems if not managed properly. Learning from defects helps – prevention of future problems, track improvements, improve prediction and estimation.	<p>Guidelines for new testers</p> <ul style="list-style-type: none">Testing can’t show that bugs don’t exist. An important reason for testing is to prevent defects. You can perform your tests, find and report bugs, but at no point can you guarantee that there are no bugs.It is impossible to test a program completely. Unfortunately this is not possible even with the simplest program because – the number of inputs is very large, number of outputs is very large, number of paths through the software is very large, and the specification is subjective to frequent changes.You can’t guarantee quality. As a software tester, you cannot test everything and are not responsible for the quality of the product. The main way that a tester can fail is to fail to report accurately a defect you have observed. It is important to remember that we seldom have little control over quality.Target environment and intended end user. Anticipating and testing the application in the environment user is expected to use is one of the major factors that should be considered. Also, considering if the application is a single user system or multi user system is important for demonstrating the ability for immediate readiness when necessary. The error case of Disney’s Lion King illustrates this. Disney Company released its first multimedia CD-ROM game for children, <i>The Lion King Animated Storybook</i>. It was highly promoted and the sales were huge. Soon there were reports that buyers were unable to get the software to work. It worked on a few systems – likely the ones that the Disney programmers used to create the game – but not on the most common systems that the general public used.No application is 100% bug free. It is more reasonable to recognize there are priorities, which may leave some less critical problems unsolved or unidentified. Simple case is the Intel Pentium bug. Enter the following equation into your PC’s calculator: (4195835 / 3145727) * 3145727 – 4195835. If the answer is zero, your computer is just fine. If you get anything else, you have an old Intel Pentium CPU with a floating -point division bug.Be the customer. Try to use the system as a lay user. To get a glimpse of this, get a person who has no idea of the application to use it for a while and you will be amazed to see the number of problems the person seem to come across. As you can see, there is no procedure involved. Doing this could actually cause the system to encounter an array of unexpected tests – repetition, stress, load, race etc.Build your credibility. Credibility is like quality that includes reliability, knowledge, consistency, reputation, trust, attitude and attention to detail. It is not instant but should be built over time and gives voice to the testers in the organization. Your keys to build credibility – identify your strengths and weaknesses, build good relations, demonstrate competency, and be willing to admit mistakes, re-assess and adjust.Test what you observe. It is very important that you test what you can observe and have access to. Writing creative test cases can help only when you have the opportunity to observe the results. So, assume nothing.Not all bugs you find will be fixed. Deciding which bugs will be fixed and which won’t is a risk-based decision. Several reasons why your bug might not be fixed is when there is no
---	---	--

HW2

What is a bug? Why do bugs occur?

Bug Life Cycle

Tuesday 11pm - Classroom

enough time, the bug is dismissed for a new feature, fixing it might be very risky or it may not be worth it because it occurs infrequently or has a work around where the user can prevent or avoid the bug. Making a wrong decision can be disastrous.

- **Review competitive products.** Gaining a good insight into various products of the same kind and getting to know their functionality and general behavior will help you design different test cases and to understand the strengths and weaknesses of your application. This will also enable you to add value and suggest new features and enhancements to your product.
- **Follow standards and processes.** As a tester, your need to conform to the standards and guidelines set by the organization. These standards pertain to reporting hierarchy, coding, documentation, testing, reporting bugs, using automated tools etc.