**BA/DA TRAINING**

**Assignment – 5(July 11,2019)**

**SHAFALI GUPTA**

1. **What do you understand by SRS document?**

**Ans:-** A software requirements specification (SRS) includes in-depth descriptions of the software that will be developed. A software requirements specification is the basis for your entire project. It lays the framework that every team involved in development will follow. It is a document or set of documentation that describes the features and behavior of a system or software application. It includes a variety of elements that attempts to define the intended functionality required by the customer to satisfy their different users.

It’s used to provide critical information to multiple teams - development, quality assurance, operations, and maintenance. This keeps everyone on the same page. Writing an SRS can also minimize overall development time and costs. Embedded development teams especially benefit from using an SRS.

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform.

SRS should include a description of the functional requirements, system requirements, technical requirements, constraints, assumptions and acceptance criteria.

**Business Drivers** - This describes the reasons why the customer is looking to build the system. Documentation which clearly identifies the business reasons for the system will help sustain support for a project if the original sponsor moves on. The drivers may include both **problems** (reasons why the current systems/processes are not sufficient) and **opportunities** (new business models that the system will make available).

**Business Model** - This section describes the underlying business model of the customer that the system will need to support. This includes such items as the organizational context, current-state and future-state diagrams, business context, key business functions and process flow diagrams. This section is usually created during the [functional analysis](http://www.inflectra.com/Ideas/Topic/Functional-Analysis.aspx) phase.

**Functional and System Requirements** - This section usually consists of a hierarchical organization of requirements, with the business/functional requirements at the highest-level and the detailed system requirements listed as their child items. Generally, the requirements are written as statements such as "System needs the ability to do x" with supporting detail and information included as necessary.

**Business and System Use Cases** - This section usually consists of a UML [use case diagram](http://www.inflectra.com/Ideas/Topic/Use-Cases.aspx) that illustrates the main external entities that will be interacting with the system together with the different use cases (objectives) that they will need to carry out. For each use-case there will be formal definition of the steps that need to be carried out to perform the business objective, together with any necessary pre-conditions and post-conditions. The business use cases are usually derived from the functional requirements and the system use cases are usually derived from the system requirements.

**Technical Requirements** - This section is used to list any of the "non-functional" requirements that essentially embody the technical environment that the product needs to operate in and include the technical constraints that it needs to operate under. These technical requirements are critical in determining how the higher-level functional requirements will get decomposed into the more specific system requirements.

**System Qualities** - This section is used to describe the "non-functional" requirements that define the "quality" of the system. These items are often known as the "-ilities" because most of them end in "ility". They included such items as: reliability, availability, serviceability, security, scalability, maintainability. Unlike the functional requirements (which are usually narrative in form), the system qualities usually consist of tables of specific metrics that the system must meet to be accepted.

**Constraints and Assumptions** - This section will outline any design constraints that have been imposed on the design of the system by the customer, thereby removing certain options from being considered by the developers. Also, this section will contain any assumptions that have been made by the requirements engineering team when gathering and analyzing the requirements. If any of the assumptions are found to be false, the system requirements specification would need to be re-evaluated to make sure that the documented requirements are still valid.

**Acceptance Criteria** - This section will describe the criteria by which the customer will "sign-off" on the final system. Depending on the methodology, this may happen at the end of the testing and quality assurance phase, or in an agile methodology, at the end of each iteration. The criteria will usually refer to the need to complete all user acceptance tests and the rectification of all defects/bugs that meet a pre-determined priority or severity threshold.

Here are five steps you can follow to write an effective SRS document.

1. Create an Outline (Or Use an SRS Template)

2. Start with a Purpose

3. Give an Overview of What You’ll Build

4. Detail Your Specific Requirements

### 5. Approval for the SRS

1. **Swot analysis for any product**

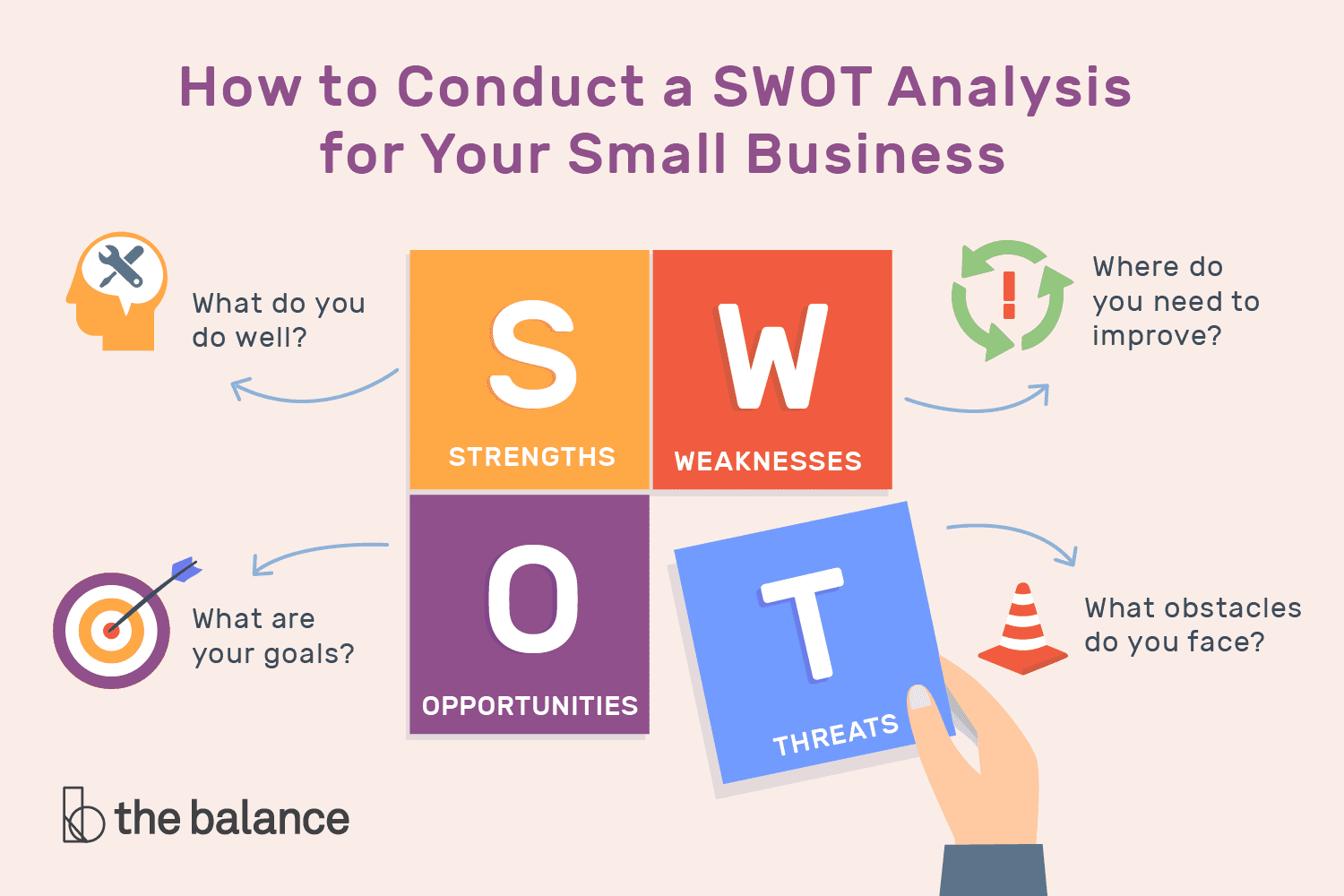
**Ans: - A** SWOT analysis is the process of documenting the internal **S**trength, **W**eaknesses, **E**xternal Opportunities and **T**hreats to your business or product. The SWOT analysis information helps to achieve the business goals. Similarly, SWOT analysis helps while launching a new product to address the four marketing verticals like **P**rice, **P**roduct**, P**romotion and **P**lace. For any product below is the exercise that needs to be done to do the SWOT analysis.

**Appraise your Strength:**

Your new product should be built around two concepts: satisfying the need or demand of a specific target audience and doing so with a unique selling benefit. This requires conducting focus groups and surveys of potential customers and a thorough examination of your competition. Your strengths might include your price, perceived value, customer service, unique features, online or retail store availability or a warranty.

**Understand your Weaknesses:**

A weakness related to a new product launch does not necessarily mean you have done something wrong – it might just signal that your competition has an advantage you have to overcome. For example, until you gain market share, you might need to spend more on marketing and give away more free samples. You might need to sell your product at break-even or at a loss for a short time to introduce yourself to customers. If your competition has negotiated exclusive endorsements and distribution agreements, that will further weaken your position.



**Identify Your Opportunities:**

Because you have a new product, you have some built-in opportunities. Early adopters and influencers like to be the first to try the hottest new thing, telling the masses who follow the lead of these groups their experience with a new product. This allows you to set your prices higher for a short period if competition is not an issue. If you have lower overhead costs than a mature competitor does, you might be able to price yourself more competitively. Take advantage of the media’s interest in new products by mounting an aggressive public relations campaign that educates news outlets.

**Assess the Threats:**

Once you enter the marketplace, your competitors will likely react. One of the biggest threats you face is your competition changing the playing field after you launch. Have backup plans for your pricing, promotion and distribution channels so you can respond to competitor changes. Consider launching in a few test markets to see how your competitors respond before you roll out your entire marketing and distribution campaigns.