Kimanuka Derrick S17B23/312

* 1. A Statement of Work is a document mainly related to project management. It is conducted between a vendor and client prior to software development. This document covers all details and expectations of collaboration.
  2. There is so much that can go wrong when making a project. These risks include:
     + The risk of having work lost:

This can happen when a laptop is stolen or crashes or attacked by viruses. Mitigating this can be done through backing up work on a physical hardrive, flash, other PC etc. or involving in services such as git and github that bring in the invaluable abilities of version control and hosting projects on their servers.

Regular committing of my project is a must-do

* + - The risk of not meeting deadlines

The problem with time is that it is short when you don’t want it to be so. Meeting deadlines and schedules is something really important in the deployment of software. When my supervisor does not clear me for a particular stage then that’s a red flag and the final deployment shall have major issues and holes. Tackling this problem is more of a personal thing and is not generic.

* + - The risk of either having too big or too small a scope

So with the making of a project, one has to make sure to get the scope just right. One shouldn’t get too carried away and make a project that Is too big for a single person. Then again it shouldn’t be too small either. This is a risk in that deadlines can be missed and huge spikes in cortisol levels for when the project is too big whereas “come back next year with a better and bigger project next time” for when the scope is small. Having a well planned out system and gathering requirements exhaustively is a way to go around this risk. Be prepared before jumping into coding.

* + - The risk of bugs and most especially last-minute bugs

Bugs are regular occurrences to software devs. We spend hours trying to fix issues in our codes. Having these issues come in last minute though is a nightmare. This can be mitigated I think through the use of version control. Having a version of mine that was working prior is something that I should commit to github and a downgrade so to speak can help me get away with one or help in finding what it is exactly that went wrong.

* + - Low productivity

It is so easy to have unproductive days – almost had one today – nowadays with all the wormholes one can get lost in. Something needs to be done everyday, or at least almost everyday in order for deadlines to be met. Mitigating this can be done through making gantt charts and keeping focused I think. Stick to your gantts I say.

* 1. Every project has stages till manifestation. My project shall follow the SDLC that many projects before mine have used. It goes as follows:
     + Planning phase: in this phase, I shall figure out in detail why exactly my system should be built and how I shall go about building it. Figuring out what business value this system can generate and the drafting of feasibility analyses all fit into this phase. This shall help me figure out whether this system of mine really makes sense to make in monetary terms anyway.
     + Analysis phase: in this phase, questions such as who will use the system, what shall it do and where and when shall it be used shall be answered. I shall investigate any current systems, identify improvement opportunities and develop a new concept for the system. Steps including the analysis strategy, requirements gathering and system proposal are done during this phase.
     + Design phase: this is where questions such as the hardware, software and network infrastructure that shall be used. In this I shall develop the database as well and what language shall be used and framework that i shall use as well if at all.
     + Implementation phase: this is where my system shall actually be built. The steps shall include ***system construction, installation*** and a ***support plan*** for maintenance and small patches as well as major changes in the future.
  2. I think my project shall be product driven due to other systems that have been done like the way mine is apiring to be. The difference between objective driven and product driven is that objective driven is a sense or picture that a developer or team has of what the final product shall be and development starts with an aim to reach to that final product. This can bring about hits and misses, where devs think to themselves – we are really off our mark.

On the other hand product driven is where devs take an already completed product and use said product as an inspiration or as a basis.

I.

According to <https://www.atlassian.com/agile>, Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

The different agile methods include the following:

* + **Scrum:** is a lightweight Agile framework that can be used by project managers to control all types of iterative and incremental projects. In Scrum, the product owner creates a product backlog that allows them to work with their team to identify and prioritize system functionality.

Note: Backlog is an accumulation of uncompleted work or matters needing to be dealt with.

* + **Lean software development:** Another iterative method that places a focus on using effective value stream mapping, to ensure the team delivers value to the customer. It is flexible and evolving; it does not have rigid guidelines or rules. The Lean method uses the following primary principles:
    - Increasing learning
    - Empowering the team
    - Fostering integrity
    - Removing waste
    - Understanding the whole
    - Making decisions as late as possible
    - Delivering the product as fast as possible

Note: Value-stream mapping, also known as "material- and information-flow mapping", is a lean-management method for analyzing the current state and designing a future state for the series of events that take a product or service from the beginning of the specific process until it reaches the customer

* + ***Extreme Programming:***

According to “https://www.xpand-it.com/blog/top-5-agile-methodologies/“, This is a typical Agile development framework, developed by Kent Beck, and can be adapted to development companies of various dimensions.

It is based around the idea of discovering “the simplest thing that will work” without putting too much weight on the long-term product view.

It is a methodology that emphasizes values such as **Communication, Simplicity, Feedback, Courage and Respect**, and prioritizes customer satisfaction over everything else. This methodology encourages trust by motivating developers to accept changes in customer requirements, even if they arrive during the latter stages of the development cycle.

* + ***Crystal:***

This is a family of Agile methodologies, and **Crystal is one of the most flexible frameworks, giving tremendous freedom to the team to develop their own processes.** It focuses way more on individuals and how they interact rather than on the process or the tools – so communication is an essential key aspect.

Crystal has variants such as **Crystal Clear** (up to an 8-person team), **Crystal Yellow** (up to a 10 to 20-person team), **Crystal Orange** (up to a 20 to 50-person team) and **Crystal Red** (for big teams with 50 to 1000 people). Crystal focuses on principles such as People, Interactions, Community, Skills, Talent and Communication, aiming to deliver the best possible software development process. The core of this development process is interaction and symbiosis, which have to exist between the people allocated to the projects and processes in order to bring efficiency to the project.

Each project is unique and undergoes frequent changes, so the team must find their own ways to bring it to its conclusion using the best decisions.

According to its founder, Alistair Cockburn, “Crystal is a family of software development methodologies, which works with the power invested by people, and is extremely light and stretch-to-fit”. Basically, Cockburn believes that **talent and the way team members interact brings benefits for the whole project.**

It’s a light methodology in terms of documentation, where teams can find their own ways over preferred work modalities, removing management overheads and creating a “free” process.

* + ***Kanban:***

The word **Kanban** is of Japanese origin and its meaning is linked to the concept of “just in time”. In practice, the Kanban method is organised on a board or table (Kanban board), divided into columns, showing every flow within the software production project. As the development evolves, the information contained in the table changes, and whenever a new task comes into play, a new “card” is created.

This methodology is also useful in individual business departments, such as HR, marketing, etc., bringing the desired visibility over all the team’s tasks.

The Kanban method **requires communication and transparency** so that the members of any team all know exactly what stage development is at and can see the status of a project at any time. It primarily focused on team capacity and is best for processes that undergo small changes.

ii.

1. There are different contracts in software projects that all depend on the seller's needs, place in life and other parameters. The contracts include:
   * **Fixed Price Contracts**

With fixed price contracts, also known as lump sum contracts, the buyer and service provider agree on a fixed price for the services in question. This type of contract is low-risk for the buyer, but high-risk for the seller since the time and costs of the project could exceed the fixed price. For this reason, a fixed price contract should include a detailed scope of work that clearly outlines what the buyer can expect for the agreed-upon price. The different types of Fixed Price Contracts include the following: ***firm fixed Price contracts, Fixed Price Incentive fee contracts, Fixed Price Award fee contracts, Fixed Price with Economic Price Adjustments etc.***

* + ***Cost Reimbursable Contracts***

When the scope of a project is unclear or subject to change, you should consider a cost reimbursable contract. This document, sometimes called cost disbursable, is also useful when the risk of a specific project is high. The seller provides work for a fixed time period or project, then increases the bill to create profit after finishing the work.

The amount of profit in this type of contract is often based on performance metrics detailed in the document itself.

The downside of this type of contract lies with the buyer, who carries the risk for this type of contract since he or she pays all costs.

Types of this contract include: ***Cost + percentage of costs/cost + fee, Cost + fixed fee, Cost + incentive fee, Cost + award fee etc.***

* + **Unit Price Contracts**

This type of contract, also called an hourly rate contract, combines elements of fixed price and cost contracts. A unit price contract pays a specified hourly rate for every hour spent on the project. It is commonly used by freelancer workers.

* + **Time and Materials Contract:**

This contract is used when labor is the main deliverable and typically provides the seller an hourly rate.