By Maurice Humado

SOFTWARE ENGINEERING

Assignments

ASSIGNMENT 1: Three (3) examples of Software Failures In Ghana

1. Port: The ICUMS (Integrated Customs Management System) introduced to accept the imports declarations, was not functioning, and leaving importers in the dark as to when they could clear their goods. **The system was not able to meet its required specification** as it kept failing at specified tasks thus causing cost problems **(budget)** for both the company and customers. The system was previously smoothly run by (Ghana Community Network Services) GCNet and operated by West Blue Consulting and was expected to be replaced by this new system.

2. The Power Distribution Services (PDS) App: The PDS launched an App through which users were supposed to pay for their electricity. This however **did not meet required specification** as users complained and cried after they had lost money trying to use that app. A renowned TV reporter also lot money via this app when she tried to pay for electricity. She rushed to a PDS office to meet one of their officers who advised her not to use that app because he didn't believe in that since it was a pilot. Her money however could not be refunded. This caused a lot of cost problems **(budget)** for the company and customers.

3. The Ghana Water Company Limited (GWCL) with SOFTtribe: Ghana Water Company Limited terminated its contract with SOFTtribe Limited, a software developing company due to what it was termed as "inefficient and ineffective service delivery". Firstly, they had been a drain **(budget)** on company resources and Secondly, after they had worked together to develop tools for meter reading and consumer application, SOFTtribe could not work up to expectation as supplied bills were inconsistent with bills generated by GWCL. (**did not meet required specification**)

Three (3) Examples of Software Failures In the World

1. Facebook's User data Leak: Last year (2019), Facebook whose ability to handle the private information had been already questioned, confirmed that nearly 50 million accounts could be at risk. Hackers exploited vulnerability in the system that allowed them to get access to accounts and possibly to the personal information of Facebook's users. In this case, I believe it also **did not meet required specification** since one of its policies was to handle and keep private information safe. I believe this will also add some **cost** to the company's expenses with regards to solving and re-enforcing the safety measures of their platform.

2. Boeing 737: Boeing discovered a software that verifies whether monitors tracking key systems of the aircraft are operating properly. This comes following a release of internal documents showing employees knew about the problems (**not meeting required specification**). Boeing employees said the aircraft was designed by clowns who in turns= are supervised by monkeys in an apparent reference to colleagues and management. Aviation Authorities across the globe have grounded the 737 Max after plane crashes in Ethiopia and Indonesia Killing 346 people. (**not meeting required specification**) Investigators have blamed an automatic control system malfunctioning during takeoff. (**not meeting required specification**). All these have also created **cost budget** for the company.

3. British Airways "Technical Issue": British Airways reported an IT system issue that resulted in the **delay** of Hundreds of flights in the UK, while dozens of flights were canceled completely. This failure affected three British airports and thousand of passengers who had to rebook their flights or check-in by using manual systems.(**not meeting required specification**)

References

https://www.ghanaweb.com/GhanaHomePage/business/UIPASS-ICUMS-not-working-Freight-forwarders-933133

https://www.modernghana.com/news/932905/pds-app-scams-tv3-journalist.html
https://www.ghanaiantimes.com.gh/gwc-terminates-contract-with-softtribe/
http://www.dzone.com/articles/the-biggest-software-failures-in-recent-years.
http://www..amp/s/amp.dw.com/en/boeing-discoveries—new-software-problem-in-737-max/a-52048332

ASSIGNMENT 2. <u>Two (2) Software Myths Under Customer, Developer And Management</u>

<u>Developer:</u>

1. “Let’s write the code, so we’ll be done faster” – This is one myth that often misleads developers. They always seem to think that the sooner you start writing your code, the sooner you’ll finish. It is never that easy. The reality is that About 60%-80% of effort is spent after first delivery to the customer, just like you spend about 60% of your time sharpening your knife when cutting a tree. This means that sooner you begin writing code, the longer it’ll take to finish.

2. “Until I finish it, I cannot assess its quality”- Another myth that often misleads developers. It is imperative for us to understand that though developers are in much of a hurry to complete coding and test, they have to pause every now and then to review their work. They are humans and make mistakes especially the ones they are not aware of. Software and design reviews are more effective than testing. The reality is that technical review is a quality filter that can be used to find certain classes of software defects from the inception of a project. Review and you’ll find 5 times more bugs.

<u>Management:</u>

1. “We can outsource it” – At times, managers overlook the fact their they have to get software developers to develop a software that fit their

specification and by that it means, knowing what you want and having the exact same thing being done for you. They overlook this fact because of time constraints and turn to outsourcing. Reality is that, if you do not know how to manage and control it internally, you will struggle to do this with that of outsiders.

2. "I buy my people the newest and state-of-the-art development tools" – It takes much more than the latest computers to do high-quality software development. Computer-aided software engineering tools are more important than hardware for achieving good quality and productivity, yet reality is that majority of software developers still do not use them effectively.

Customers:

1. "We can refine the requirements later" – More often, customers embrace a notion that a personal statement of objectives is sufficient to begin writing programs and that they can fill in the details later. But this is a rather a recipe for disaster. This poor thinking is the major cause of failed software efforts. A formal and detailed description, including the functions, performance, behavior, interface, design, information domain and validation criteria is essential and these characteristics however are determined only after there is thorough communication between the developer and the customer.

References

https://www2.cs.siu.edu/~mengxia/Courses%20PPT/435/Chapter_01.pdf -

ASSIGNMENT 3: The 5ws and 1 H of Software Engineering

The 5ws and 1 H stands for the What, Why, When, Where, Who and How.

5Ws stand for What, Why, When, Where, and Who. 1H stands for How

What: Thinking of What initiates the process of understanding the basics of issue, problem or scenario at hand. It is about cognitive mapping of the scope of the issue, problem, or scenario.

Why: Asking 'Why' entails clarifying why the issue, problem or situation at hand occurred. It aims to identify the triggers and rationalizes the occurrence of an issue or a problem.

When: This element is about time-stamping the occurrence of an issue or a problem. Having an understanding of the time of occurrence could help in sequencing the triggers and impact of the issue, or problem.

Where: This element is used to pinpoint the location or place of occurrence and hence could be helpful in identifying the people and other things present / existing at that location which may have contributed to the occurrence of an issue or problem.

Who: This is about identifying people who may have direct or indirect involvement in causing or contributing to the issue or problem.

How: This element of technique is used to examine the sequence of things and triggers and how the resultant problem or issue unfolded.

References

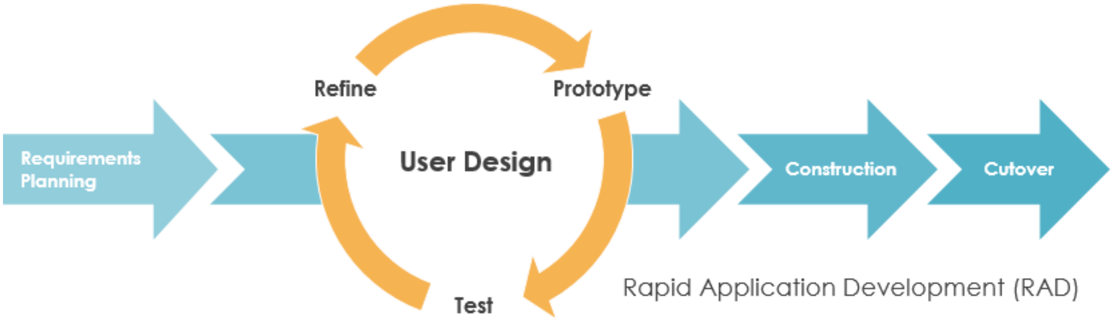
https://en.wikipedia.org/wiki/Five_W

ASSIGNMENT 4 <u>Two Software Process Models</u>

- RAD model

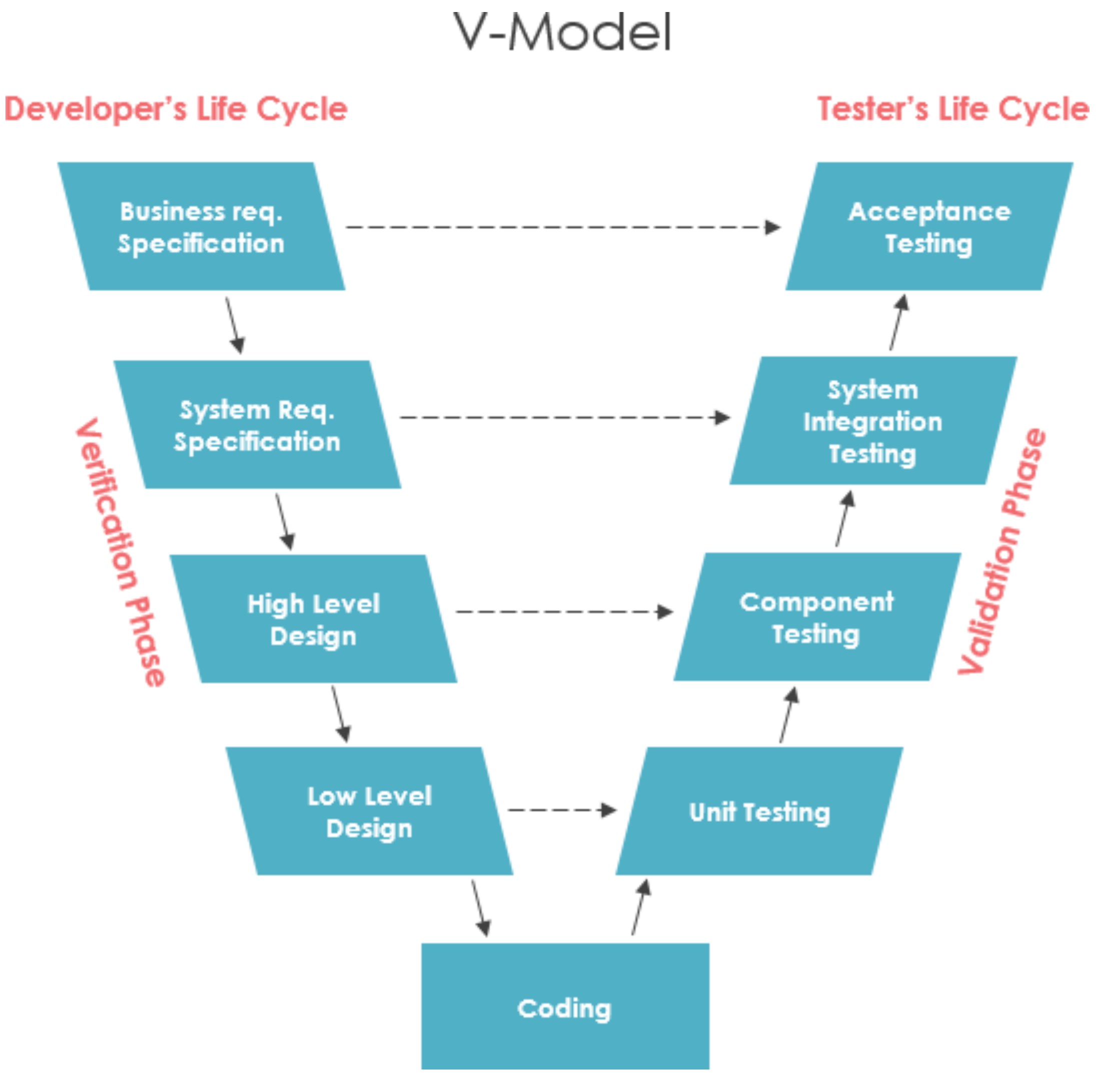
Rapid application development (RAD) is often referred as the adaptive software development. RAD is an incremental prototyping approach to software development that end users can produce better feedback when examining a live system, as opposed to working strictly with documentation. It puts less emphasis on planning and more emphasis on an adaptive process.

RAD may resulted in a lower level of rejection when the application is placed into production, but this success most often comes at the expense of a dramatic overruns in project costs and schedule. RAD approach is especially well suited for developing software that is driven by user interface requirements. Thus, some GUI builders are often called rapid application development tools.



- The V-Model

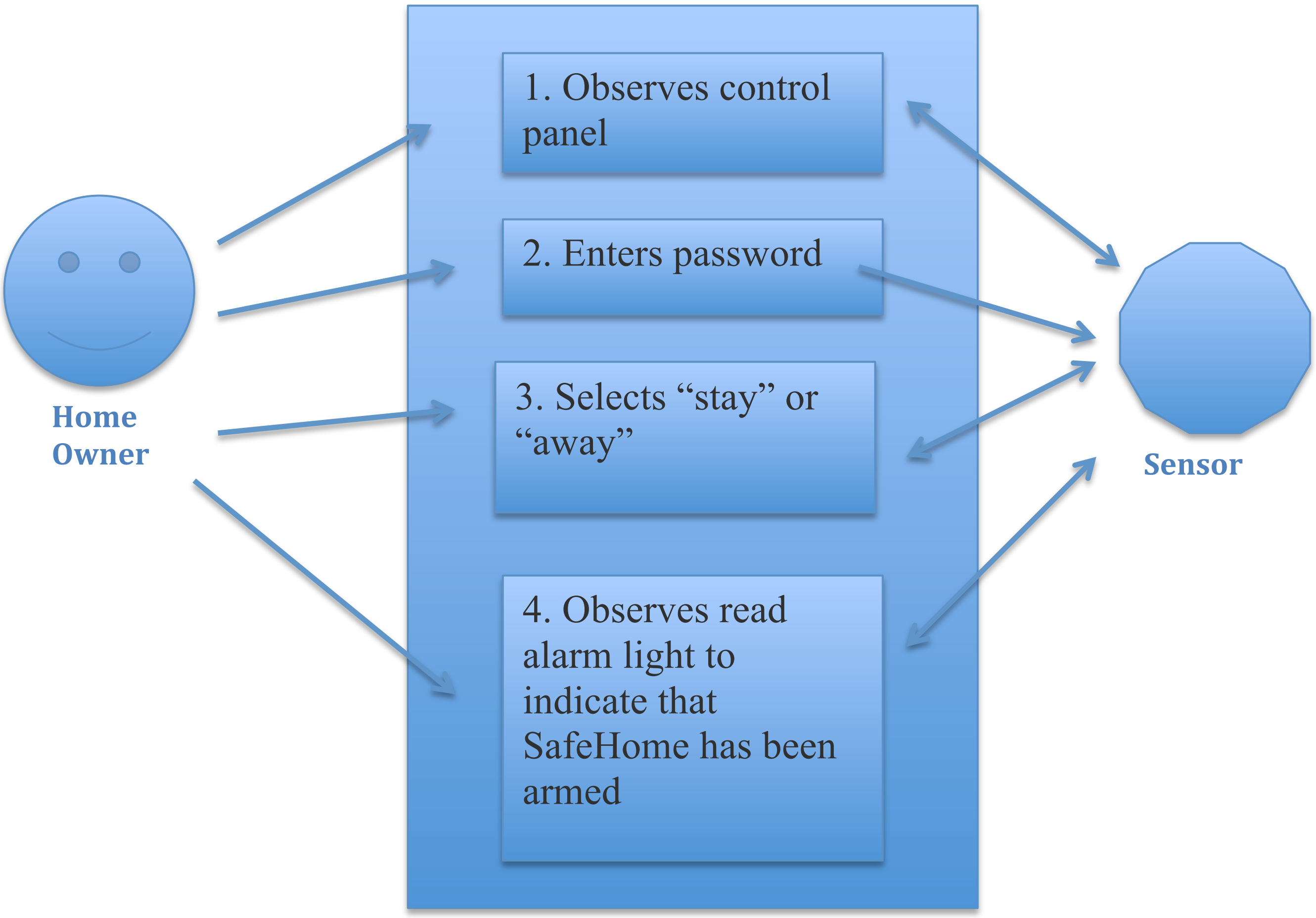
With the V-model, Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. The horizontal and vertical axes represent time or project completeness (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively.



References

https://www.visual-paradigm.com/guide/software-development-process/what-is-a-software-process-model/

ASSIGNMENT 5: DESIGN THE UML OF THE HOME OWNER


1. Observes control panel
2. Enters password
3. Selects “stay” or “away”
4. Observes read alarm light to indicate that SafeHome has been armed
Home Owner
Sensor

Monitoring and response subsystem

Main Actors

Homeowner - a user

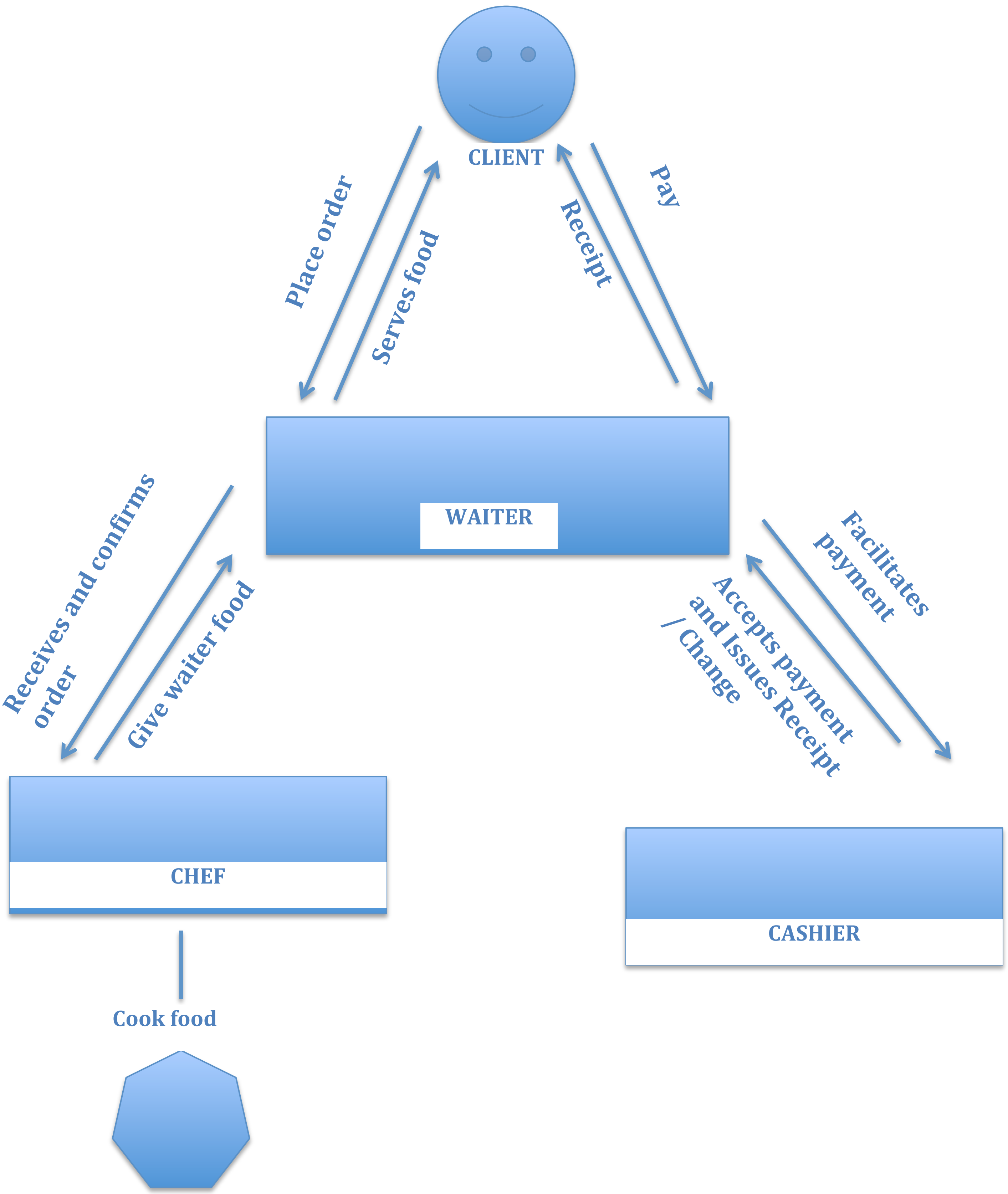
Sensors - devices attached to the system

Monitoring and response subsystem - the central station that monitors the SafeHome home security function

Scenario:

1. Homeowner: observes control panel
2. Homeowner: enters password
3. Homeowner: selects “stay” or “away”
4. Homeowner: observes read alarm light to indicate that SafeHome has been armed

ASSIGNMENT 6: FLOW DIAGRAM OF FOOD/DRINKS DELIVERY IN A RESTAURANT



ASSIGNMENT 7: Main Deliverables/ Artifacts of Software Engineering Process Life Cycle.

A deliverable is a tangible or intangible good or service produced as a result of a project that is intended to be delivered to a customer (either internal or external). A deliverable could be a report, a document, a software product, a server upgrade or any other building block of an overall project. The Product /Project Manager oversees these deliverables. The following are the main deliverables of through a life cycle of the Engineering Process or Project

1. Project Definition Document: This is mostly done by product manager who comes up with brings the creativity, ideas, etc. after the project or product is stated or defined. This document is mainly based on the scope, time and budget of the product.

2. Business Requirement Document: This is done by the Business Analysts. After, the project definition document is ready, business Analysts get the information, does their work and analysis and also comes up with this document. This is mostly based on mockups, wireframes, prototypes, which is often done with the help of other teams depending on how they are structured.

3. Functional Specification Document: This is done by the System Analysts who also gets the information from the business analysts so as to be able to come up with a design that identifies all the different measurements and specification. This mostly includes the Text specifications or System specifications.

4. Software Development System (Raw System): The product engineers, software developers or programmers are responsible for the deliverable. They try to develop the basic or raw system before the touch ups and testing are done.

5. System: (For Testing) The system is created and tested with their testing tools and signed off and passed off for production.

6. Production or Live System: After testing, the system is ready and produced after the market and customers or client world.

7. Update: This stage mostly has to do with fixing of bugs and defect troubleshooting and a lot of updates per requests and challenges faced from the system after a period of time.

ASSIGNMENT 8: HOSPITAL APPOINTMENT MANAGEMENT SYSTEM

Crystal Uni Hospital, Adjei Kodjo.

Requirements;

Main Actors;

1. Front Desk Operator (Receptionist)
2. Patient
3. Doctor
4. Administrator

Functional Requirements:

There are a lot of software requirements specifications included in the functional requirements of the Hospital Management System, which contains various process, namely Registration, Check out, Report Generation, and Database.

Requirement;

Registration Process

● Adding Patients: The system enables the staff in the front desk to include new patients to the system.

● Assigning an ID to the patients: The system enables the staff in the front desk to provide a unique ID for each patient and then add them to the record sheet of the patient. The patients can utilize the ID throughout their hospital stay.

Check Out:

● Deleting Patient ID: The staff in the administration section of the ward can delete the patient ID(with regards to admission or stay) from the system when the patient's checkout from the hospital.

● Adding to beds available list: The Staff in the administration section of the ward can put the bed empty in the list of beds-available.

Report Generation:

● Information of the Patient: The System generates a report on every patient regarding various information like patients name, Phone number, bed number, the doctor's name whom its assigns, ward name, and more.

● Availability of the Bed: The system also helps in generating reports on the availability of the bed regarding the information like bed number unoccupied or occupied, ward name, and more.

Database:

● Mandatory Patient Information: Every patient has some necessary data like phone number, their first and last name, personal health number, postal code, country, address, city, 'patient's ID number, etc.

● Updating information of the Patient: The system enables users to update the information of the patient as described in the mandatory information included.

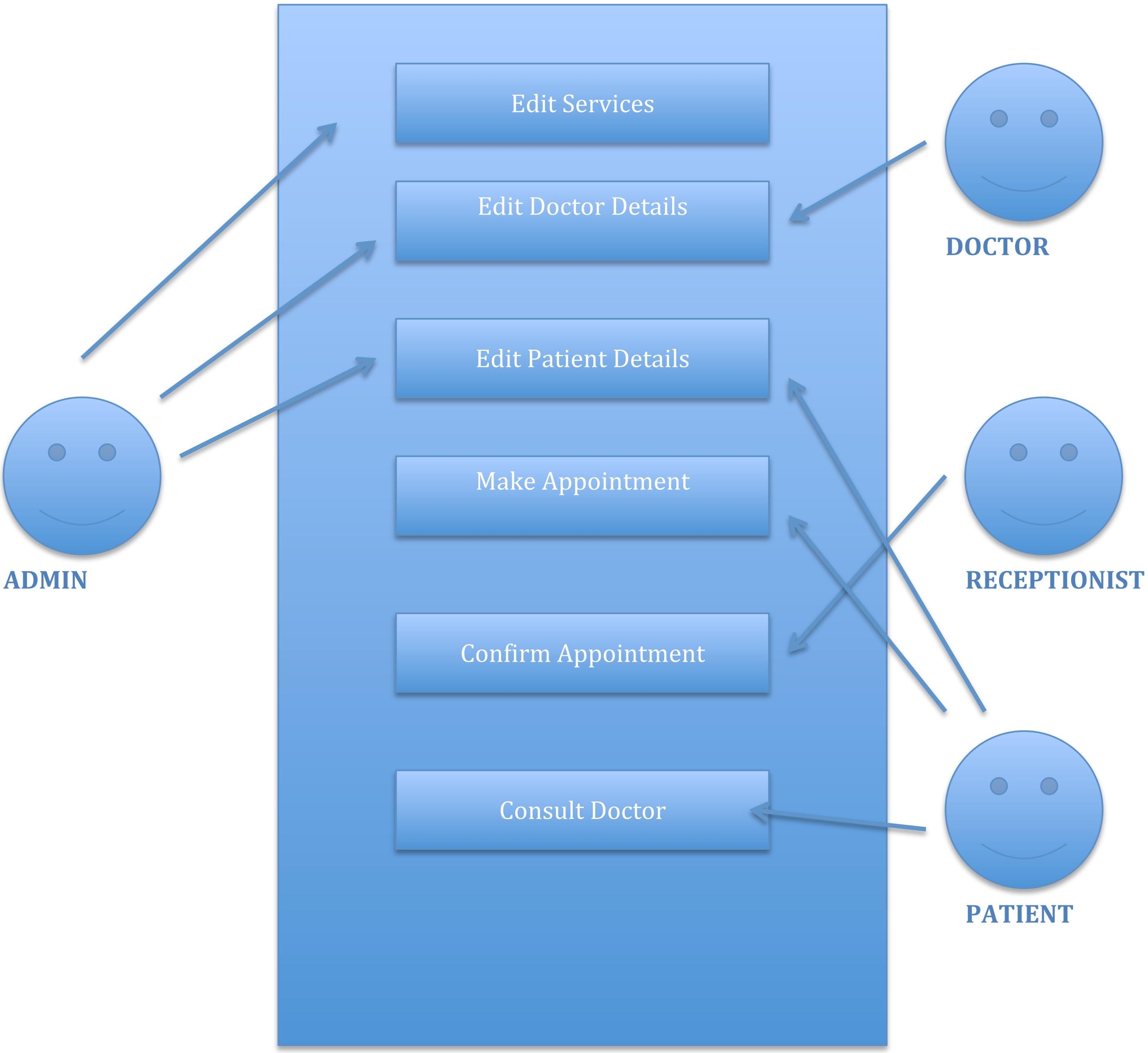
Scenario (From Requirement Gathering);

1. If new patient, personal details are taken and registered onto the system. Details include; Name, Age, Sex, Marital Status, Contact, Next of Kin.
2. i. If call, Structure with Available Specialists with Time Slots. E.g. Patient A to see Doctor B between an available time e.g. 10am-11am.

   ii. If walk in, structure with their available specialists with time slots or general doctor without time slots but mindful of time given to caller to be reserved or managed.
3. If specialist doctors aren't available, time period given to walk and call patients.

This section contains all the users' functional requirements. Each requirement is prioritized with M-Mandatory, D- Desirable and O-Optional as follows:

ITEM	REQUIREMENT	TYPE
1.1	The software should generate a unique identification number for every patient.	M
1.2	Admin should be able to search patient details as well as edit details.	M
1.3	Should be able to provide a structure with doctors and their available time slots	M
1.4	The software should keep a record of patient file and diagnosis as well as doctor consulted at the time.	M
1.5	The software should show that time admitted and time discharged with regards to bed capacity and details.	M
1.6	The software should allow a self booking and with any doctor as the user wants as well as cancel appointment	D
1.7	Should access all previous sheds-in details and medical past appointments	O
1.8	Should show doctor details and time availability	M


Edit Services
Edit Doctor Details
Edit Patient Details
Make Appointment
Confirm Appointment
Consult Doctor
ADMIN
DOCTOR
RECEPTIONIST
PATIENT

ASSIGNMENT 9: REQUIREMENT AND SPECIFICATION OF A SALON/SPA BOOKING APPOINTMENT SYSTEM

A beauty salon or spa is an establishment dealing with cosmetic and relaxation treatments for men and women.

Objectives:

- Eliminate usage of diaries to note down appointment details, writing manual invoices for the payments done by the Customers etc.
- Eliminate the data redundancy; keeping appointment details at several places (Diary, mobile etc.) by several people (Owner, Employees, Customers etc.).
- Efficient and effectiveness of the Salon management activities, services and processes like maintaining Customers, Employees, Appointments and Payments etc.
- Ease the management and decision making while improving Salons' good name.
- Client satisfaction and Employee satisfaction.

Scope:

- Providing the facility to registering regular Customers and maintaining their details.
- Facilitate appointment handling.
- View appointments, leaves, and holidays through an event calendar.
- Handling Salon Services along with their respective prices, hours etc.
- Providing Customer Payment handling option.
- Generating invoices through the system.
- Generating reports to support the higher managerial decisions.
- Maintaining an information centre (dashboard).
- A Reminder-generating facility through emails

Requirements

Basically, Non-functional requirements describe 'how the system works', while functional requirements describe 'what the system should do'. Observations and Interviews were conducted as facts gathering methods at the requirement-gathering phase in order to gather requirements.

Functional Requirements

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used.

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail.

- Create regular customers and maintaining (Update/ Inactive) their details.
- Create salon services and maintaining (Update/ Delete) their details such as prices, hours etc.
- Maintain resources (Create/Update/ Delete) at the salon premise.
- Reminder generating facility in the system and send via mails for all the respective stakeholders.
- Maintain holidays and staff leaves.
- Facilitate appointment handling through an event calendar by the system.
- Providing customer payment handling option.
- Enter payment details for the system.
- View customers' payment balance details.
- View customers' gross payment details.
- Generating invoices through the system.
- Generating reports to support the higher managerial decisions.
- Maintaining an information centre to display crucial data charts.

Non-Functional Requirements

Basically, non-functional requirements relate to qualities of the system that cut across user facing features, such as security, reliability, and performance [5].

- Accessibility – The system is able to be access anywhere at any time by the authorized users.
- Accuracy – The correctness of data inputs to the system was ensured.
- Availability – System is available within working hours. But can be used at special occasions also. E.g.: - At a bridal dressing
- Maintainability – This is a considerable factor especially for a non-technical user.
- Privacy –Confidentiality of data inputs to the system has been assured.
- Reliability - Ability of the suggested system to function under conditions for a specified period of time has been assured.
- Security – The data feeds to the system has been protected by controlling the user access privileges.

Main Actors

1. System administrator
2. Stylist/ Employee
3. Cashier

System Administrator

- Allow creating Employees/ System Users.
- Allow creating regular Clients.
- Allow generating crucial Reports.
- Allow viewing crucial Reports.
- Allow viewing Information Centre.
- Allow viewing payment information.
- Allow viewing appointments, leaves and holidays calendar.

Stylist/ Other Employee

- Allow creating appointments.
- Allow maintain appointment details.

- Allow maintaining appointment statuses.
- Allow viewing appointments, leaves and holidays calendar.

Cashier

- Allow making payments
- Allow printing invoices.
- Allow viewing appointments, leaves and holidays calendar for the Customer perspective accordingly.

All the relevant designed documents are displayed at this section.

These are for the case of Salon Management System for 'Salon Nirosha'.

This table describes Client Registration.

Use Case:	Client Registration
Actors:	System Officer
Description:	Register the regular clients of the salon.
Pre-Conditions:	
System Officer should login to the system. System Officer should be able to access the Client Registration module.	
Flow of Events:	
1. Select 'Profile Management' menu. 2. Select 'Client Registration' menu. 3. Fill the details respectively. 4. Click on the 'Register' button; Or else click on the 'Cancel' button.	

This table describes Appointment Creation.

Use Case:	Appointment Creation
Actors:	System Officer
Description:	Create Appointments.
Pre-Conditions:	
System Officer should login to the system.	
Flow of Events:	
1. Select 'Appointment Management' menu. 2. Select 'Appointment Creation menu. 3. Fill the details respectively. 4. Click on the 'Create button; Or else click on the 'Clear button.	

This table describes Appointment Reschedule.

Use Case:	Appointment Reschedule
Actors:	System Officer
Description:	Reschedule Appointments.
Pre-Conditions:	
System Officer should login to the system.	
Flow of Events:	
1. Select ‘Appointment Management’ menu. 2. Select ‘Appointment Reschedule’ menu. 3. Edit the details respectively. 4. Click on the ‘Reschedule’ button; Or else click on the ‘Cancel button.	

This table describes Appointment Status Management.

Use Case:	Appointment Status Management.
Actors:	System Officer
Description:	Change Statuses of the created Appointments time to time and cancelling the appointments.
Pre-Conditions:	
System Officer should login to the system.	
Flow of Events:	
1. Select ‘Appointment Management’ menu. 2. Select ‘Appointment Status’ menu. 3. Select respective option button. 4. If selected the ‘cancel’ button: 1. Enter the reason for cancellation. 2. Click on the ‘Save’ button.	

Other than the specified users for each use case, the System Administrator has been given the access privileges to create/ view/ update/ delete respective data at the total system.

Reference:

https://pdfs.semanticscholar.org/b50d/1a39b9b65912268ae0e65a64bad87c1ad184.pdf

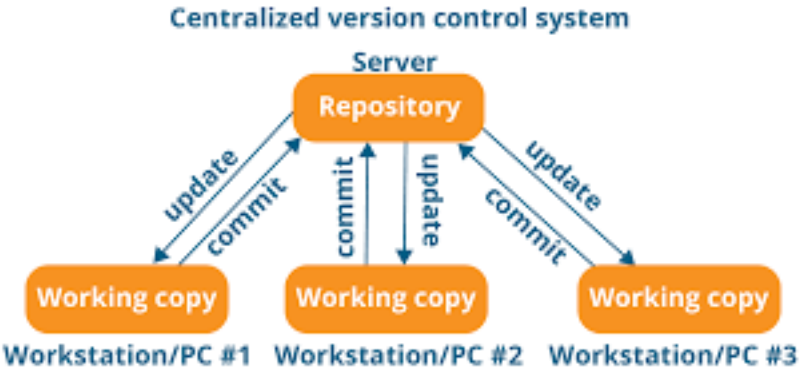
ASSIGNMENT 10: GIVE TWO (2) MAIN VERSION CONTROL SYSTEMS IN SOFTWARE DEVELEOPEMENT.

Version Control Systems are process management systems which maintain changes recorded in a file or set of files over period of time. Each change is maintained as a version. Users can track specific versions later. The unique features of version control system are as follows: Up to date history is available for the document and file types; It does not require any other repository systems. The repositories can be cloned as per the need and availability. This is extremely helpful in case of failure and accidental deletions.

There are three (3) main types of version control system.
They are the **Local Version Control System**; **Centralized Version Control System;** and **Distributed Version Control System.** Lets delve into 2 of them.

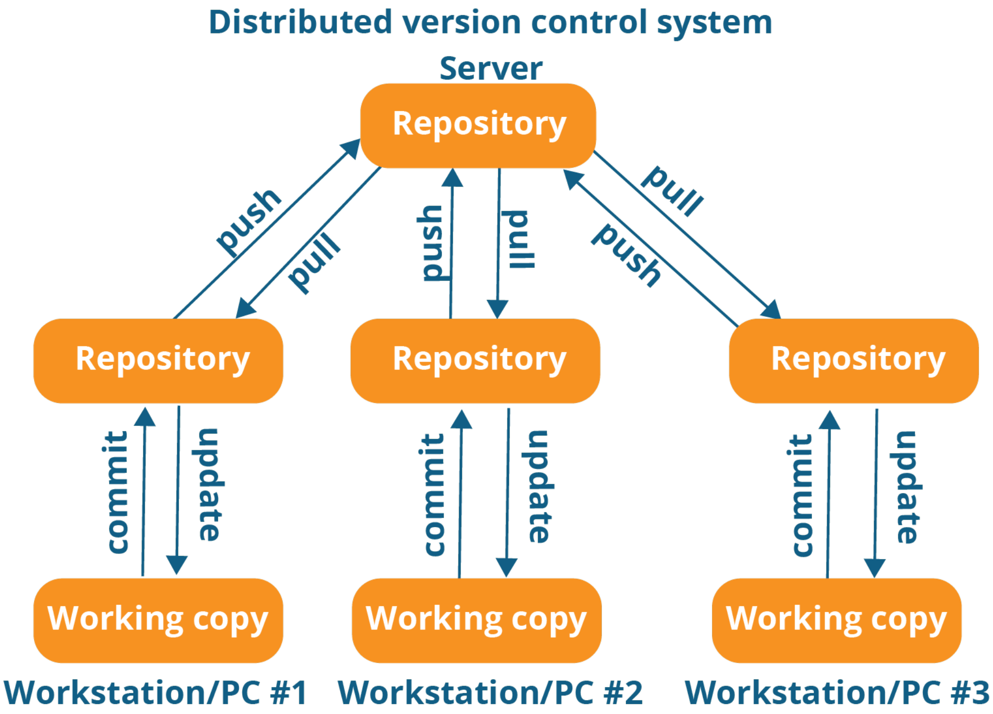
1. Centralized Version Control System:

In this system, all the changes in the files are tracked under the centralized server. The centralized server includes all the information of versioned files, and list of clients that check out files from that central place. An example is the Tortoise **SVN**- "**Apache Subversion**", a popular version control system tool.



2. Distributed Version Control System:

Distributed version control systems come into picture to overcome the drawback of centralized version control system. The clients completely clone the repository including its full history. If any server dies, any of the client repositories can be copied on to the server, which helps restore the server. Every clone is considered as a full backup of all the data. An example is the **Git**, which has an emphasis on speed and performance. It is supported by all operating systems.



References:

https://blog.eduonix.com/software-development/learn-three-types-version-control-systems/

ASSIGNMENT 11: DIFFERENCE BETWEEN VERIFICATION AND VALIDATION

The terms Verification and Validation are commonly used in software engineering to mean two different types of analysis.

Validation- asks the question "Are we building the right system"?

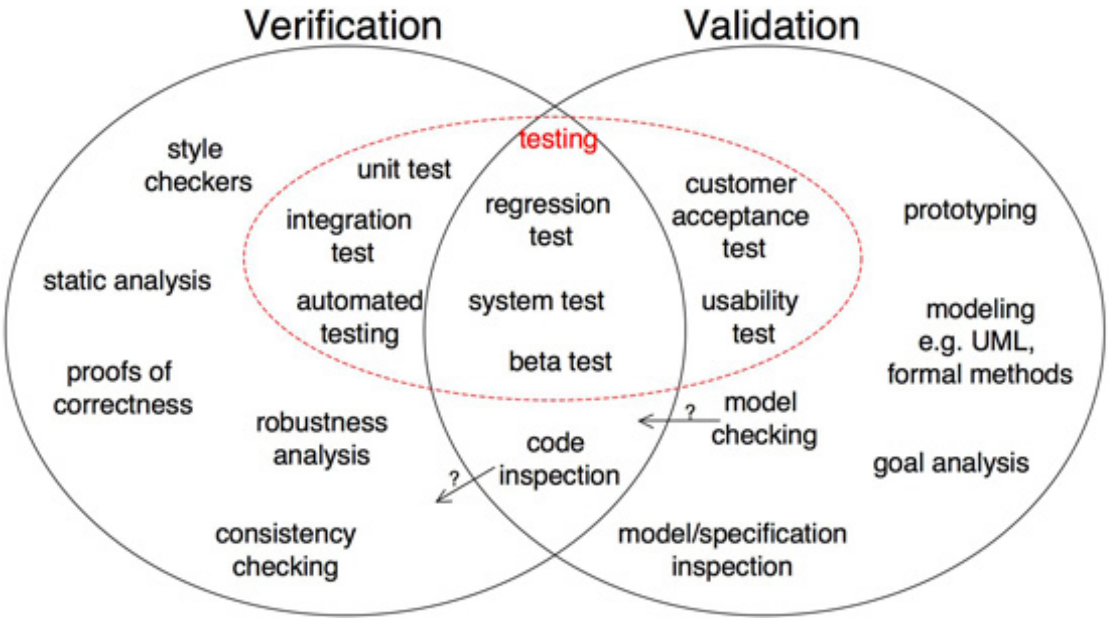
Verification- asks the question "Are we building the system right"?

In other words, validation is concerned with checking that the system will meet the customer's actual needs, while verification is concerned with whether the system is well-engineered, error-free, and so on. Verification will help to determine whether the software is of high quality, but it will not ensure that the system is useful.

The distinction between the two terms is largely to do with the role of specifications. Validation is the process of checking whether the specification captures the customer's needs, while verification is the process of checking that the software meets the specification.

Verification includes all the activities associated with the producing high quality software: testing, inspection, design analysis, specification analysis, and so on.

Validation includes activities such as requirements modelling, prototyping and user evaluation.


Verification
Validation
style checkers
static analysis
proofs of correctness
robustness analysis
consistency checking
testing
unit test
integration test
automated testing
regression test
system test
beta test
customer acceptance test
usability test
code inspection
?
?
model checking
model/specification inspection
prototyping
modeling e.g. UML, formal methods
goal analysis