

NOTE: Exam study is providing examples + Past exam papers + drawing relationship

<https://www.youtube.com/watch?v=uLUpDUqVeZY> ← exam question in this case

## ICT393Ans1

Approaches (incl. formal methodology & broader approaches) to system development-

**Buying software solutions approach** involves purchasing the required software solutions. Often considered when there is a lack of in-house personnel to maintain or develop the required software solution. However, often there a downside such as there may be a slow turnaround or additional for software modification. The software solution may not be able to achieve all the requirements or changing requirements. And may require on going prescription payments and the intellectual property may not remain in the organisation.

**Building software solutions approach** involves personnel in the organisation are in charge of developing the software solution the organisation requires. The advantage is if there are available resources such as personnel it may be more cost efficient also the software solution can be modified to suit the changing organisation requirements. The disadvantage is the organisation are dependent on the skills of the available personnel to develop and perhaps if the personnel leave the organisations, it may be difficult for others to maintain the existing software solution. Also, may disadvantage the organisation in that critical resources are reallocated to develop the software.

### Characteristics of system development methodology/approaches

**Prescriptive approach** is where all the planned-out steps for system development are adhered to. Suitable for when the system development can be actually planned in advance and requirements are clear and of low risk of changing.

**Adaptive approach** is where there is flexibility in having to follow the steps for system development. Meaning the development may adapt to changing requirements within guidelines. Suitable when requirements are not clear and at high risk of changing often because of high technical risk involved.

For example, in scrum the product owner is able to change or prioritise certain client requirements to be developed. Also, the client is able to make modifications to his existing requirements. A prescriptive approach would mean the original high-level requirements need to be adhered to and changes to it are discouraged. This makes Scrum inappropriate as being prescriptive

**\*Sequential approach** is where the (system) development involves developing the system in phases where it is completed in linear fashion and no overlap or iteration. The output of each phase is the input of the next phase thus a delay in previous phase will delay the start of the next phase. The advantage is that requirements are defined clearly and progress can be tracked. Reduces the likelihood of changing requirements during development because of less stakeholder interaction. The disadvantage is if one phase gets delayed it delay the next phase thus could extend deadline also high risk meaning the end deliverable may not reflect what he wants. For example- Waterfall methodology.

**\*Iterative/Incremental/Spiral approach** is where the (system) development involves developing the complex system in iterations where each iteration involves all of the core processes. This approach allows for continual refinement of the system and encourages feedback and interaction from stakeholders through showing them each iteration result. Also risk of failure is lessened because changes can be requested at end of iteration so expectation is met. The disadvantage is scope is not known in advance. For example- Agile methodology.

For example, in Scrum a single sprint contains planning, building, testing and reviewing. A scrum project has multiple sprints. Thus, it would be more appropriate to be classified as iterative. Since with iterative approach projects planning is normally all done at the start of a project. A scrum project planning is done throughout and even in each sprint. Additionally, a project classified as having an iterative approach to development would have testing as its own phase while in Scrum testing is done throughout the entire project.

**Model driven approach** is where the (system) development involves the emphasis of creating models to aid in visualisation and analysis of problems, requirements, and design systems. The models that are potentially involved include- Object modelling (UML), Data modelling, and process modelling.

The advantages could be productivity in that models makes the engineering process easier to understand for all parties. The models promote communication between different parties such as the project manager and developer prior to the development of system. This means the requirements and deliverables are clear to all parties involved. Another advantage is that it reduces the risk of changing personnel negatively impacting the development and maintenance of the system. The new personnel are able to get a high-level overview of the system and reduce the risk of misunderstanding the system from the created models. Third advantage could be assisted in the delegation of appropriate skill level to appropriate tasks.

The disadvantage could be the speed at which the system is developed is not the quickest. Meaning the time taken to create the models and discuss them could have been used on development. This could anger the stakeholders who may need it urgently. Another disadvantage could be it requires more resources to implement effectively. To take full advantage often the collaboration between the different parties needs to occur and in smaller organisation the different parties may be busy or there may be there at all.

**Product driven approach** is where the (system) development involves the emphasis of rapidly creating the prototype of the system or developing the code rapidly. This aims to provide stakeholders with a minimal viable solution as soon as possible.

For example, Scrum is a product driven approach because after each sprint an improved minimum viable product is created

The advantage is that it increases the speed at which the stakeholders get a prototype product meaning if the stakeholder decides to cancel request, then the time and cost spent on the product will be minimal. Another advantage is the non-technical stakeholders or clients can be involved in development process. The minimal viable product provided can be

evaluated and allows the stakeholders to provide the next steps of improvements thus providing some control to the stakeholder. The disadvantage could be increased complexity. When the prototype is shown to stakeholder the stakeholder may ask for a complex customisation or change the requirements. This could result in having to say no to the stakeholder. Another disadvantage is it could lead to poor communication with business and developers meaning the requirement of the business are not fulfilled by the developers because of lack of high-level communication at start.

## System Development Methodologies

**System Development Methodologies** refers to the framework that is used to plan, structure, and guide the development of system development. It defines techniques, activities, automated tools, and deliverables for developing a system. These methodologies are adopted for use by project managers and system developers in an organization. The appropriate methodology for a particular project maximizes the efficiency at which a system is developed and success rate of a system development. There is various methodology for different scenarios in an organization. For example, in an organization that seeks These include- waterfall and agile approach

### Factors impacting methodology choice:

The product may impact approach. An appropriate approach must match the product. A simple system that can be clearly defined and planned out can use less formal approach since it's often quickest to develop. Also, a more critical system would use a more formal approach to reduce mistakes

For example- missile guidance system or system used in military would use a more formal approach since mistakes or errors could result in the loss of mass human lives

The deadline of development. Given a short deadline for developing a system an iterative approach may not be best since stakeholder interaction may slow down development.

For example- Every year a new FIFA game must be released. Thus, we need to predict the timeline of the project to ensure it meets the year deadline. So an iterative approach may be appropriate

The cost. For projects that are high in cost to develop a formal methodology would be suitable. An iterative approach could be used since the iteration reduce the risk of failure in the end product since adjustments can be made

For example-

## System Development Modelling Techniques

**System development techniques** is the individual techniques used to support system development. For example, various modelling techniques.

**Modelling techniques** are ways to create a model. It is individual techniques used to support system development. UML (unified modelling language) is the default technique that contains different diagrams to represent different aspects of a system. Components UML components include-

## UML Use Case Diagram

**Use case diagram (UML)** shows the relationship between the actors and use case and the internal and external factors of the system. The actor is named the end user or client. The use case is the task the system must be able to perform a response to. The system boundary or automation boundary separates what's inside system and outside generally actors who use system. One way for the business analyst to confirm requirements with stakeholder through a simple diagram that non-technical users can understand and follow. Essentially bridging gap between IT and business with client expectation managed and hopefully met. They help with drawing business process models by helping identify actors which could be lane in business process model. Also helps identify the use cases and their associated actors, which is relevant to business process models since the use case can be considered activities

## UML System Sequence Diagram

**System sequence diagram (UML)** illustrates the sequence of messages flows passed between user and the system for a use case or functionality. The components consist of system, user and input and output messaging. This is can be used to ensure the developers implement the business requirements. Sequence diagrams are useful for communicating how a system operates for a use case. Also, useful to ensure that if there is a new developer, they will have an understanding on how to fulfill the user requirement. Which is what the business analyst is responsible for. This information is useful for business process models because it helps identify roles in business process through actors in system sequence diagram.

## UML Activity Diagram [TODO]

**Activity diagram** models the sequences of activities done between the actors and the system in order to achieve a use case. The activity diagram serves as a useful way to communicate the procedures and process to the stakeholders. Helps the business analyst with gathering and evaluation business requirements from clients

Don't have certain symbols like time and artifacts

**State machine diagram** is used to illustrate an objects different possible behaviors or states and how and object transitions to the different behaviors or states. Capture how events can trigger change in behavior of the object. The components of state machine diagram consist of state which is the behavior of an object based on some condition and transition which represents a new state for the object. It helps simply communicate to the client the different behavior an object may have. This ensure that the client is not surprised by an unexpected functionality or behavior. Also allows for the client to confirm or reject certain behavior

## UML Class Diagram

**UML class diagram** illustrates the entire high-level structure of the system in terms of the classes, relationships, operations, and attributes. They provide assistance to business analyst in that they are able to clarify and verify the scope of the system and the defined functionality before implementation. It also helps provide the blueprint for developers to implement the system in a way that ensures the functional requirements are met. The

specific types of class diagrams include domain model class diagram and the design class diagram. The domain model class diagram is more involved in helping clarify and verify the scope of the system and the defined functionality before implementation. The design class diagram is more involved in providing the blueprint for developers because it contains greater details such as relationship type, methods and properties. This information helps with business process modelling by helping validate the sequence flow of the business process model. In the class diagram it highlights different relationships. If two classes in UML class diagram don't have a relationship with each other than in the business process model any sequence flows between the business process model equivalent class needs to be reviewed. It helps highlight potentially logic errors in relationships

**Domain class model/diagram** are used to summaries the information requirement of the system from user perspective. Produced from requirement analysis. Provides the business analyst a way to deeper understanding of the problem domain and to resolve unclear user requirements.

**Design class model/diagram** are produced from activities of system design. They represent specifications for the application interface and classes.

**Entity relationship diagram** illustrates how people, object and concepts relate to each other in a system. Most commonly used in design and debug of databases. Used to model a particular database to be implemented. Help communicate the landscape to different teams and business. Enables ability to remove unnecessary process or ambiguities within organization

TODO: Draw connections + packaged software

## ICT393Ans2

### Traditional System Development VS Agile Development

**Traditional system development** refers to system development methodology that adopt a sequential approach. This is where in the development cycle one phase only begins after the full completion of the previous phase. So, the incompleteness of one of the previous phases would halt the start of another phase. Traditional system development is prescriptive in nature meaning the planned-out steps for system development are adhered to. This allows for more accurate cost and time projections and prediction of a project. Model driven approach in nature where the (system) development involves the emphasis of creating models to aid in visualisation and analysis of problems, requirements, and design systems. Since being able to determine the requirements at the start is a necessity and drawing models helps support this.

**Waterfall development** refers to system development methodology that adopt a sequential approach, where in the development cycle it is done in phases one after the other where if the incompleteness of one phase halts the start of another. The phases often include- Requirement analysis, development and testing, and deployment.

**Agile development** refers to system development methodologies that adopt an iterative approach to development, where in a development cycle there are regular releases of an improved minimum viable product. In agile development collaboration is emphasised; the collaboration between different self-organised, cross-functional teams ends up spurring the evolution of the requirements and solutions. Agile development is adaptive in nature meaning changes even at the last minute are embraced [changes could be different prioritisation]. The continual embrace of changes and regular releases of the working deliverable means less emphasis is placed on detailed documentation and more on timely delivery of high-quality software. Also agile is product driven approach in nature where in scrum after each sprint an improved minimum viable product is created.

### Issues with Agile System Development

#### Issues/Problems with agile system development-

Projects that have development teams or resources spread across different geographical locations is not ideal for agile development. Having teams in different geographical locations unable to easily and constantly have face to face communication violates one of the principles in the agile manifesto. This violated principle describes how face to face conversation provides the most effective and efficient way to exchange information. In an agile development environment regular communication between teams is done. Not easily allowing face to face communication increases the likelihood of information not being communicated or misinterpreted which increases the chances of project failure, especially in an agile environment where regular communication is needed.

For example- if one team was in a location in a different time zone then the time zone difference could reduce the availability of each party to communicate. One team could be sleeping while the other team is wanting to chat.

<https://www.spkaa.com/blog/the-12-principles-behind-the-agile-manifesto-principle-number-six-promote-face-to-face-interaction>

Projects that outsource some of their activities is not suited for agile development. Companies that carry out outsourced activities generally are set up to gather all the requirements of activity and carry out the activity. Almost no room is provided for changes to the agreed upon requirements. In agile development changing requirements is expected and encouraged. Thus, the difficulty in getting the outsourced activities to be modified or improved based on changing requirements makes project success less likely. One solution could be finding a service provider that understands agile development and allows for changing requirements

<https://www.iauro.com/is-agile-development-effective-when-outsourcing-software-development>

Projects that involve the creation of failsafe systems is not suited for agile development. The nature of agile development revolves around the idea of fast releases of an improved program after every couple of weeks. So, with an agile approach small issues are normally ignored in order to stick to the release schedules. Over time these small issues build up and could leave the final overall system more vulnerable to failures or attacks. Also, if the developers leave then the lack of documentation may make future maintenance of failsafe system and extension works challenging.

Projects that involve the creation of large software systems is often not suited for agile development. The agile nature means less time is spent on the big picture design. More emphasis is placed on implementing the solution and refactoring it as the project progress along. Since less time is spent on the big picture design the software systems built is more difficult to change or modify. Larger software systems tend to require the most changes. Thus, this approach may not be the most suitable

Projects that involve many large teams is not suited for agile development. Having a larger team means increased need to communicate information to even more people. As teams become larger it becomes easier for information not to be passed through or misinterpreted when passed down. With agile development approach the risk is amplified due the increased communication.

One problem with using agile development is that the items that are built are generally not reusable in different context. The code being written are done quickly to solve specific immediate issues. Those items were not built with the idea of being reused. This could negatively impact the overall benefit of agile approach adoption. There may be a scenario where an organisation is asked to build a different variation of a past project. Had reusability been at the forefront future similar projects could be delivered even faster. Thus, agile adoption may hinder the efficiency of future projects being built

Characteristics of agile development include: (not reworded because these are general and doesn't make it distinct compared to traditional development. Traditional development can encourage teamwork, accountable etc)

A project management process that encourages frequent inspection and adaptation

A leadership philosophy that encourages teamwork, self-organisation and accountability

Development practices that allow for rapid delivery of high-quality software

## A business approach that aligns development with customer needs and company goals

Agile development methodologies include:

### Scrum

**Scrum** is a type of agile development methodologies. It adopts the agile development principles. It provides a framework for teams to work together to develop a product with greater emphasis on embracing requirement changes and customer involvement during the process. It is an iterative process meaning the product development is split into iterations called sprints where an improved minimum viable product is created. The duration allocated to each sprint is between 1 to 3 weeks. It relies on user requirement prioritisation for each sprint. A product backlog is used to help facilitate this.

Scrum process are as follows-

The **product backlog** is a list of all high-level requirements ordered in terms of importance. The product owner creates the product backlog with information provided by the client. The product owner passes the product backlog to the sprint team during the sprint planning event. The sprint team are self-organising and consist of developers that carry out the sprints. They will review the product backlog and select the items to be completed in the sprint. The items will be outlined by the sprint team in the sprint backlog. The sprint backlog must be able to be completed within the allocated sprint duration. The sprint backlog is then used as the foundation for the sprint. A **sprint** is a work process in scrum that delivers at the end of each sprint an improved minimum viable product for the client. Each sprint has defined time duration. The sprint aims to implement specific user requirement(s) from the sprint backlog. Each sprint consists of- planning, building, testing and reviewing. Also, each day of the sprint there is a scrum meeting to report progress. After the sprint event the sprint review event occurs. This is generally when deliverable of the sprint is demonstrated to the client and feedback is provided. Finally, the sprint retrospective event occurs this is when the team reflects on the sprint process efficiency.

Components of scrum-

**Client** is actively involved in the process and will review the sprint deliverables and provide any additional suggestions to the product backlog.

**Product owner** responsible for maintaining the product backlog list. He is the main point of communication with the client and will make changes to the product backlog as per the client's requests.

**Scrum team** are self-organising and consist of developers that carry out the sprints. This includes determine the requirement priorities for each sprint.



**Sprint** is a work process in scrum that delivers at the end of each sprint an improved minimum viable product for the client. Each sprint has defined time duration. The sprint aims to implement a specific user requirement(s). Each sprint consists of- planning, building, testing and reviewing. Also

### Parts of a sprint

**Scrum master** is the project facilitator

### Extreme programming (XP)

**Extreme programming (XP)** contains values and practises that when followed in the extreme programming process reduces the risks of development process and lead to high quality results. It also helps teams work faster and be able to collaborate more better. There is a greater emphasis on technical aspect to the software development since the principles detail how to best implement. Pair programming occurs to encourage communication and increase code quality. There is a large emphasis on testing and customer is responsible for providing feature prioritisation. This order outlined by customer is normally strictly followed. The frequent iterations are done to release minimum viable product to customer for feedback and to keep customer involved.

The process of extreme programming is as followed and iterate continuously-

**Planning** is the stage where user stories are created by customers to outline the system features in priority. The team creates a release plan consisting of all the iterations each coverings a desired system feature.

**Designing** is the stage where underlines the importance of simple designs

**Coding** is the phase where the feature is implemented and extreme programming practise are followed these include- pair programming, code refactoring, continuous integration, code ownership and coding standards

Pair programming occurs to encourage communication and increase code quality. There is a large emphasis on testing and customer is responsible for providing feature prioritisation. This order outlined by customer is normally strictly followed. The frequent iterations are done to release minimum viable product to customer for feedback and to keep customer involved.

**Testing** contains unit testing and acceptance testing and are implemented before coding

**Listening** is where there is constant communication from all stakeholders during the process. Small frequent releases are provided to the customers after each iteration for feedback

<https://www.altexsoft.com/blog/business/extreme-programming-values-principles-and-practices/>

[https://www.tutorialspoint.com/extreme\\_programming/extreme\\_programming\\_process\\_cycle.htm](https://www.tutorialspoint.com/extreme_programming/extreme_programming_process_cycle.htm)

XP has 5 values-

Communication - with open, frequent verbal discussions

Simplicity - in designing and implementing solutions

Feedback - on functionality, requirements, designs and code

Courage - in facing choices such as throwing away bad code or standing up to a too-tight schedule

Respect

XP practices-

Pair programming - 2 programmers work together on designing, coding, and testing

Simple designs - "KISS" and design continuously

Refactoring - improving code without changing what it does

Owning the code collectively - anyone can modify any piece of code

Continuous integration - small pieces of code are integrated into the system daily or more often

System metaphor - guides members towards a vision of the system

Traditional vs agile project management

#### **Agile approach project management characteristics-**

Agile approach project cost management process tends to be more challenging because by nature agile approach embraces changes to the requirements. These changes could make it difficult to predict the overall cost. Also, system is not designed upfront generally. Traditional approach tends to be better at project cost management. The cost is considered at the start and rarely changes.

Agile approach project communication management process is more critical because the nature of agile approach relies on constant communication with all stakeholders. Each release is a collaborative effort. Traditional approach put less emphasis on communication. For example, the client is not actively involved in all the SDLC stages. Normally feedback and changes is given at the end of the development.

Agile approach quality management process needs refactoring and continual testing. The nature of agile development revolves around the idea of fast releases of an improved program after every couple of weeks. So, with an agile approach small issues are normally ignored in order to stick to the release schedules. Over time these small issues build up and could leave the final overall system more vulnerable to failures or attacks. Thus, emphasis still needs to be placed on testing. Traditional approach quality management process consists of time allocated for detailed testing. This means project quality is set up to be high.

Agile approach project time management process tends to consist of a more realistic work schedule because the activities are broken down into releases and thus can be judged on the duration easier.

Agile approach scope management process has clients more actively involved with the scope. The scope tends to change because of agile nature.

## ICT393Ans3

### User Experience

**User experience** refers to the emotion individual feel when using a system. User interface can be good but user experience can still be poor. So essentially while user interface is good it doesn't do the task it intended or promised. Includes how they feel about the discovery/delivery process of system, the user interface itself through interacting with system, the whether the system delivers the features as promised. (Also, company associated with the system, and support received when issue arises with system)

For example- A new video editor offers a great user interface for video editing. However, the user experience is poor due to the video editor not allowing the editing of videos that have a duration of greater than 2 minutes

**User interface design** is ONE ASPECT of user experience design. If user interface is good then it positively impacts the user experience and improves it to an extent

### User Experience Design

**User experience design** refers to evaluating the emotions of individual when they work with the system and improving the satisfaction of user by improving how easy the system is to use, the efficiency of executing the tasks and satisfaction associated with the interaction

For example- ease of use (UI), how you feel out interaction such as services around it including ease of repairs

### Is This Good Design or Bad?

Human Factors Designers Need to Consider For Good Design / Why is this good UX design  
Attributes to consider for good or better design- (people attributes & product attributes)

**Emotion** refers to the possible emotions users might feel when using the system and how we plan to combat those emotions. Designers that is able to understand possible user reactions to the system are able to design the system to better deal with the various reaction.

For instance, when it comes to TIKTOK design certain users will feel happy when they see videos with puppies in it. Tiktok has been designed to show those users more puppy videos. On the other hand, other users may see a dog video and feel no emotion and so Tiktok will not show them the video.

For instance, young people may hate seeing videos of dog racing ..

Examples of emotions include- being afraid to use the system, frustrated when using system,

**Context** refers to looking at the environment in which the system will be used and ensuring that it's supported. Designers that are able to identify how and where a system will be used are able to design the system to better support the identified environment

For instance, the video streaming system is be able to be used without internet capabilities. Design so recent locally streamed videos are downloaded to device

For example- mine field, on the go, multiple platform

**Motivation** refers to what is driving people to use the system (or product or service) and why. Designers that are able to find the reason for the client's interest in adopting the system can better tailor the system to suit and meet the clients' requirements.

For instance, if a client only motivation in purchasing a phone from us is because our phone offers a head jack. Then the phone we are offering should contain a headphone jack

For example- mandatory for work, on sale

**Psychology** refers to information related to physical person characteristics and how system can support it or combat it with their design. Designers that can gather the physical characteristics of their clients that may impede their user experience are able to design their system to be able to better suit their client.

For instance, if their client is colour blind then the system needs an option to toggle colour blind mode on or be developed for only colour blind clients

**Goals** refers to looking at what users aim to do with system and supporting it

#### Product Specific Factors Designers Need to Consider (By Employer or Client)

**Constraint** refers to limitation placed on the design or the designer when building the system. Designers that ignore the constraints that are placed are likely to end up designing a product that offers clients a bad user experience. The constraints placed may include- include size limitation, cost limitation, technical skill limitation and development time allocated. Ignoring these constraints puts the product at risk of not delivering the features or requirements promised thus making user experience bad

For example- A requested employee management system designed ends up being more expensive to operate and maintain than the agreed upon price. This hurts the designers reputation of being reliable and may put the designed system at risk of being cancelled by client

For example- a system that needs to be designed and implemented is going to be carried about by people. If the people implementing the design are unable to or don't have the skills required then the design will not be possible. And time would have been wasted.

**Materials** refers to the understanding of the materials at which the product will be built from. Designers that don't consider the materials when designing are likely to end up designing a product that offers clients a bad user experience. The material information can include- whether vendors supply the material, the cost of the materials or whether the hardware is technically capable for running the system. Not taking this information into account puts the product at risk of not delivering the features or requirements promised thus making user experience bad

For example- if the system requires hardware to have GPU, then the need to make sure the all hardware have GPU. Also, if the screen for the system requires special OLED, then need to have a discussion with the vendors that supply it.

**Features** refers to the underlying functional requirements that the stakeholders have requested for the system (product or service). A system could be designed well and easy to use; however, if it doesn't implement the features promised then it would leave the user with a poor user experience. Designers whom actively don't attempt to provide features client requested is at risk of designing a product that provides bad user experience

For example- a university might want to implement LMS. The design could work for mobile really well and layout could be intuitive but if features such being able to watch past lectures or ability to enter an online tutorial is not implemented then it is considered poor user experience.

**Service** refers to understanding the outside or external factors possibly impacting the product. The external factors surrounding the product could be ease getting faulty product serviced or repaired, ease of getting system upgraded, or ease and transparency of purchasing the product

For example- a server company may have a control panel system to enable customers to managing their servers. The ease at which a server fault can be reported by a customer and repaired will impact user experience. If customer is unable to report the fault via a ticketing system or through calling for example then the user experience maybe poor.

## UX Designer Tasks

Skills needed by User Experience designer-

**Good research skill** refers to understanding more on good user experience by looking up articles or surveys. Then implementing the research. Also enables individuals to look at current system design and identify weakness and improvements. Important because good user experience is not static and often evolves with the time and new technologies. The method to carry our research includes-

Customer interviews which are where the potential users or clients are asked questions in relation to the system or product. This evidence will be used considered when designing system or product.

Observation is looking at how potential users or current users are interacting with the system and product

**Front end developing (not just designing)** refers to coding front end with languages like HTML and CSS. The methods that may be adopted for front end developing

User test where the GUI created is evaluated on its usability and functions. Use case test can be programmed. User testing can include checking whether the product is compatible in various requested browsers and devices

Sketching

**Good visual designer (aesthetic)** refers how things look in terms of art

**Good product strategy** refers to skills in terms of long-term strategy of design and how it will evolve to get better in the future

**Good user interface design**

Data for user experience design-

### Small data

**Small data** is data that is in a format and small data set volume that is accessible and can be comprehended by humans. People are able to understand the data set and derive conclusion(s) from it. Can be considered big data that has been collected, organised, and packaged by algorithm and machine so that it can be comprehended and understood by people. Obtained by traditional user research methods such as interviews, observing and scrapping. Small data can be easily analysed and reveal quick potential insight into customer behaviour, engagement and preferences

For instance, a restaurant owner can evaluate customer reviews of their restaurant and if waiting times for dishes to arrive are frequently raised as too long the owner can hire more staff to rectify. This improves user experience because customers are no longer angered by the excess wait time for their food to arrival

For instance- charts

### Big data

**Big data** is data that is in a format and large data set volume that can only be comprehended by machines. People are unable to understand the data and derive conclusion(s) from it without machine assistance. Has a variety of structured or unstructured formats including text, images, audio and video. Big data can be analysed and provide insight into customer behaviour, engagement and preferences.

For instance, Netflix uses data such as users viewing day, time, device, location, and browsing patterns and performs big data analytics. This reveal viewing trends of user and indicates whether users are happy about the recommendations provided to them by Netflix. This insight enables development of improved movie suggestions resulting in better user experience

For example- sensors gather weather data over country and computer process data and transform into a chart (small data)

For example- all the news published by users on facebook

Category	Big Data	Small Data
Data Sources	Data generated outside the enterprise from nontraditional data sources, include: <ul style="list-style-type: none"> <li>• Social media</li> <li>• Sensor data</li> <li>• Log data</li> <li>• Device data</li> <li>• Video, Images, ect.</li> </ul>	Traditional enterprise data. Includes: <ul style="list-style-type: none"> <li>• Enterprise Resource Planning transactional data</li> <li>• Customer Relationship Management (CRM) system</li> <li>• Web transactions</li> <li>• Financial data e.g. general ledger data</li> </ul>
Volume	<ul style="list-style-type: none"> <li>• Terrabytes (10<sup>12</sup>)</li> <li>• Petabytes (10<sup>15</sup>)</li> <li>• Exabytes (10<sup>18</sup>)</li> <li>• Zettabytes(10<sup>21</sup>)</li> </ul>	<ul style="list-style-type: none"> <li>• Gigabytes (10<sup>9</sup>)</li> <li>• Terabytes (10<sup>12</sup>)</li> </ul>
		• Batch or near real time

### Behind the scenes

Being a large enterprise, Netflix deals with massive amounts of data from its over 150 million subscribers. With the streaming industry becoming more and more competitive, the subscription-based company uses all this information to their advantage to offer targeted experiences to their customers. According to Data And Analytics Network, the data they collect includes:

- Viewing day, time, device, and location
- Keywords and number of searches
- The number of times you paused, rewound, fast-forwarded, and rewatched a content
- Browsing and scrolling patterns
- And even how much time a user takes to finish a movie or a TV show

By applying a series of algorithms to the massive amounts of customer data they possess, Netflix is able to predict what the user will watch next, but also offer a range of options based on the aforementioned data. This method has proven to be very successful for Netflix, as 80% of the content being steamed is based on their recommendations algorithm.

<https://www.youtube.com/watch?v=D2U9KHWUrFw>

Method/Process for user experience and interface design-

**Traditional design process** is sequential process for the design. There is generally a separation between design team and development team. When the design team finishes their duties, such as researching and providing a layout outline, they hand it over to the development team to implement. The design team often has no future responsibilities. The development team is now responsible for implementing or making design changes along the way. This may be an issue because the small designs decisions made by development team could negatively impact the user experience. Had the design team stayed and saw out the implementation they could have pointed it out with their knowledge.

**Agile design process** is iterative process for the design. Its emphasis a cross functional team

## ICT393Ans4

**Business process** is a string of tasks, decisions, events involving several actors and objects that triggered by a reason, and leading to a result that has customer value. Can be classified as either a primary process or support process. (Think to do one thing)

For example- text book has many businesses process in that you have some for printing, delivering

For example- student context have many business processes such as enrolment, fee paying, getting books from library

### Business Process Type

Classification of business processes-

**Primary (core) processes** are where the process is initiated due to outside processes. Often deals with the process of delivery or producing of business product/services/deliverables to customer. They are considered the essential activities of the organisation that need to be performed to meet its goals, objectives, and mission

For example- server company marketing/servicing/managing/deploying servers are primary process

**Support processes** are processes that enable conditions for primary processes to work. These processes aren't involved in directly producing or delivering business product/services/deliverables

For example- server company produces/delivers servers to client. A support process would be the (payroll) process of getting server workers paid where it doesn't directly contribute to server company producing/delivering servers to clients

For example- Procurement (buying things in), human resource management and firm infrastructure

(Think- Do the process contribute to with our service/product of the company. Whatever company is in business for is primary process)

Basic business process for business receiving order + dispatching good (where business processes outlines TASKS only)

**Business processing modelling** is a way of representing the organisational processes using models. Business process models explains how a business functions in terms of how the activities or tasks are accomplished. A single model is generally used to describe a single business process unless it provides a high-level view of many processes. \*Look at it from an instance. Does it represent different cases\*



## Advantages of Business Process Modelling

### Importance/benefit of Business Process Modelling-

An importance/benefit of business process modelling is that it assists with software development in that it provides a simplified way to communicate different complex processes and provides an easy way for others to understand the complex process. This is beneficial for programmer looking to implement the complex process in their application.

Another importance/benefit of business process modelling is that it helps with training/new starters. The documentation is given to new starters to understand how things are done. Beneficial for keeping organisational knowledge

Another importance/benefit of business process modelling is that it assists with the restructuring of business processes so faster/cheaper/more efficient. Need to know how they work first.

## Issues with Existing Business Process [Q1]

Issues related to business process could be associated with-

Costs is described as ...

**Productivity** is described as the measure of how well resources in an organisation are utilised to complete a particular task

For example, how well human resources, such as the coach, uses his time

Efficiency

Service delivery

Elimination of waste (related to lean approach)

Fitness for commercial software / possible automation

Streamlining process for better collaboration with other processes or partners

Removing defects and variability (related to six sigma)

## Advantages of Business Process Restructuring

### Importance/benefit of Business Process Restructuring

Reduce costs

Improve productivity

Improve efficiency

Improve customer service (include service deliver to customer)

Fit commercial software

Streamline the supply chain (better collaboration with business partners)

~~Design for automation~~

## Business Process Reengineering vs Business Process Management

**Business process Reengineering** seeks to improve business process performance through radically changing and redesigning the entire business process from scratch. It does not offer incremental improvement to a business process but the full improvement in one go. The difficulty is that it is prone to failure often due to lack of staff adoption.

**Business Process Management** seeks to improve the business process performance through utilising principles, utilities, and methods to design, evaluate, execute and track the business process. Covers the different data, stakeholders and components such as- customers, employees, IT infrastructure, suppliers and data because it influences business process performance. Treats the existing business process as a template to be analysed and redesigned to be better rather than completely redesigning

## Business Process Management Lifecycle

**Business Process Management Lifecycle** is a diagram that illustrates how a business process is managed to ensure that the business process is operating at its most optimal performance. It provides a strategy for continually improving business process. The business process management lifecycle contains the following stages-

**Process identification** stage is where the suitable business process which we seek to improve is identified. This stage activities may include determining whether the business process is a core, support or management process and working out whether information about the business process is available.

**Process Discovery (BPM Lifecycle Phase)** deals with getting information on an existing process and utilising it to generate an As-IS process model. This is required because in the real-world details are not written down on a piece of paper line by line ready for process modelling. Real world requires the extraction of the information needed in various ways. The various tasks done at this phase including creating a team, gathering information about process, creating As-IS model and assess and ensure the model meets different quality criteria.

**Process analysis** stage is where the business process is evaluated to reveal its current and future issues, strengths and risk. This process can use the AS-IS model to serve as the basis for analysis. Also, the As-Is model helps makes it easier identify issues in the business process due to its visual representation. Potential issues in the business process can related to cost, productivity, efficiency and service delivery. A cause-and-effect diagram can be created and used to provide insight into potential causes of the issues.

**Process redesign** is the stage where the business process is redesigned in order to resolve the issues, potential risks, and weakness identified in the process analysis stage. The AS-IS model is used as the initial starting model in which the to-be model will be generated from. Meaning the business process model is not created from scratch. The To-Be model is essentially improvements to the AS-Is model. At the start of process redesign phase, the way to assess whether a redesigned business process is an improvement or not is defined. This is basically setting the business process performance indicators. For example- a devil's quadrangle which contains cost, time, and flexibility dimensions can be used as the business process performance indicator

**Process implementation** stage is where the improved business process is integrated and implemented. Recommended to deploy the improved business process on a smaller scale first to test the improved business process performance. Consider the effect of the changes on technology and people such as designing training to help them with the transition,

**Process monitoring and controlling** is the stage where the business process is tracked on its performance and business performance improvement opportunities can be identified. Data can be collected about the business process performance in order to make it easier to identify any bottlenecks or irregularities. For example, time taken to carry out the process

**Lean** is an approach that focuses on the how resources are utilised in the business process. The approach considers resources that are utilised and that don't contribute to the creation of value for the customer as targeted for elimination. Aims to reduce the resources utilised when creating more value for the customer. The end result is the improved business process should use less capital, space, human effort and money

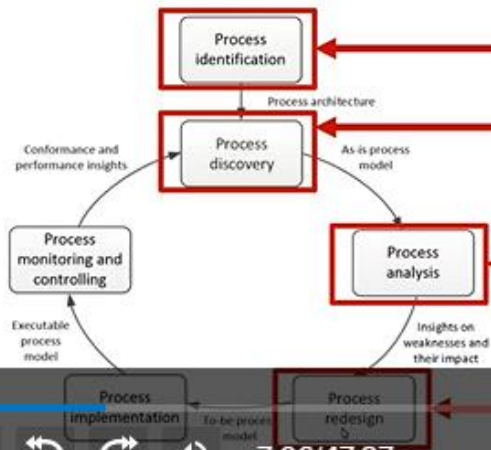
## **Six Sigma**

Six Sigma is an approach that seeks to improve the quality of process outputs by identifying and removing the causes of defects and variability in manufacturing and business processes

It uses a set of quality management methods, including statistical methods, and requires people within the organization who are experts in these methods

Each Six Sigma project carried out within an organisation follows a defined sequence of steps and has quantified financial targets (cost reduction or profit increase)

### BPM Lifecycle



### Assignment Task

- Step 1** Identify a suitable **business process** that could use some improvement
- Step 2** Gather information about the process and use that information to create an **As-is process model**
- Step 3** Analyse the process using the As-is model to identify any **issues** that need to be resolved



- Step 4** Redesign the process to include proposed improvements - i.e., create a **To-be process model**\*



7:06/47:27



1x



## ICT393Ans5 [Not done]

A BPMN process model consists of **four types of core elements**:



**BPM (modelling) N process diagrams** is standard notation for representing business processes in a graphical flowchart.

**Microsoft Visio** is a simpler application that supports the activities in business process identification, discovery, analysis and redesign. Focused on purely the modelling of business process and systems

helps business process identification, analysis,

supports the activities in

**Naming elements convention:**

**FOR Activities** use verb & noun (assess credit risk) verb start BECAUSE people/machine have to do

**FOR Events** use noun & past tense verb (insurance claim lodged) past tense verb end COS already happened

Event:

1. Start event
2. End event (must have end point to every possible thing that could happen in process cos it doesn't disappear to nowhere)

XOR Gateway: (1 path only EXCLUSIVE)

EXCLUSIVE & TAKES ONLY ONE BRANCH

1. XOR-split Comments on line
2. XOR-join



AND Gateway: (All paths happen. At the same time.)

ONLY PASS if BOTH path happen at SAME TIME

1. AND-split
2. AND-join



OR Gateway: (1 path only OR more)

1. OR-split
2. OR-join



Timer event: (Occur after x days. Indicate delay)

NOTE: Status GO ON ARROWS

Looping

Pool (Encapsulates many lanes)

Message flow (arrows connect two pools)

Lane (Inside pool models departments/internal roles/systems)

### BPMN VS Activity diagram

TIPS

- Draw *lines* for decisions (write choices on line)
  - One line is drawn easy to add symbols/gateway based on logic
- Do the *Tasks/Activities* (rectangles) based on VERB + NOUN
- Do start *event* at least + End *event* (can have multiple end events)
  
- IGNORE the actor and start with the VERB of actor. Assume actor is a lane
- Having 1 GATEWAY results in either 1 more gate way (assume) or a split
- Difficult bit is event
  - Start/End event easiest
  - Remember according to FMG: If something is sent follow up with received event
- ~~Very low level considers EVERY verb something i.e prepare and send~~

## ICT393Ans6

**Process Discovery (BPM Lifecycle Phase)** is required because in the real-world details are not written down on a piece of paper line by line ready for process modelling. Real world required to extract the information needed in various ways. This phase deals with getting information on an existing process and utilising it to generate an *As-IS process model*. The intention is to make that particular process faster/cheaper etc. ~~Done by business analyst and domain experts~~

Tasks of process discovery include-

**Assemble team** task is responsible for creating a team that will be involved in and responsible for working on process discovery

(Last three can be done iteratively and repeated not sequentially)

**Gathering information** task deals with trying to get a better understanding of the relevant process. To get a better understanding of the relevant process information needs to be gathered about the process and it can be done through different process discovery methods

**Conduct the modelling** task deals with actually creating a model of the relevant process. The use of a modelling method assists with creating the model in a systematic way

**Assure model quality** task deals with assessing the quality of the created model of the relevant process in order to ensure the model meets different quality criteria. Helps build trust in the process models

### Issues with Process Discovery

**Challenges/Problems with process discovery/interacting with domain experts include-**

One challenge is having to deal with the knowledge of the process being fragmented. The ability to pull the knowledge from various and different sources in order to model the process is the



challenge. Often process tasks will be completed by separate resources. The domain experts responsible for their resources are called on to explain their tasks. The issue is the domain experts themselves often have an abstract understanding of the overall process. A domain expert may express how a particular output of their resources is important to another upstream task. However, the domain expert of that upstream task may provide conflicting information on what outputs are important for their tasks. Thus, each domain experts may have different assumptions. So, the fragmented and conflicting information regarding the relevant process needs to be resolved. This is normally needs to be done in several iterations

For example-

Another challenge is domain experts express their tasks in instances level. Meaning when asked to describe their tasks generally it may be difficult. Since the tasks may differ in different cases. So, the questions asked to domain experts should be about what happens in specific cases to their tasks. The answers to these can then be used to abstract the process to a higher level

For example- If student enrolls without prerequisites the following tasks are? If student enrolls in an already completed unit again then the task are?

Domain experts can describe tasks in different instances but when asked to describe whole enrolment process it is unlikely or too difficult to describe

Third challenge is domain experts often are not familiar with the business process modelling language. When domain experts are asked to provide feedback on a draft process model, they are unable to read it and provide feedback. So, for business analyst it is better to explain content of model in more detail and in a natural language to the domain experts. The domain experts can then point out problems in a high-level verbal way

For example- a business analyst will describe the process verbally and the domain expert will follow along. This enables the domain expert to eventually understand and contribute where needed

**Process discovery methods** relates to the ways to gather information requirement for modelling a process. Three classes include: evidence-based, interview-based, and workshop-based discovery process

[Evidence vs Interview Vs Workshop Based Discovery \(Ways to gather info about business process\)](#)

**Evidence-based discovery** are information obtained directly from the process the information is available in different forms of evidence.

This method has low to medium level of time consumption meaning amount of time required for this discovery process is low to medium. For instance, document analysis uses existing documentation available and time is mostly taken for the analysis. However, for observation time is taken for analysis and also observing the process.

\*This method has medium level of richness meaning insights provided about the process is medium level. For instance, in documentation analysis issues identified from the analysis may arise but further information about the issue is often not explored and can't be explained in the documentation.

\*This method has high level of objectivity meaning information gathered about the process is highly unbiased.

**Document analysis** is where the information is collected from existing documentation about the relevant process. Examples of documentation can be process descriptions, internal policies, glossaries, organisation charts, forms, and handbooks and work instructions.

For example- MFL bylaws

**Observation** is where an individual case of the process is performed and followed to get an understanding of the process.

The observer can take an active role meaning they play a specific role in the process and trigger the execution of the process. The execution steps and choices are recorded but obviously from the perspective of the specific role.

For example, taking the role as a player during season registration.

The observer can take a passive role meaning observe participants of process and their environment

**Automated process discovery** is where information is collected from event logs that are relevant to the process. Event logs may capture resources that executed tasks and actual execution of the process. For example- event logs may have cost information, actors in system

**Interview-based discovery** are information obtained from interviewing domain experts about how a process is executed. Iterative in nature because knowledge about the process tends to be fragmented. Requires interviews to many domain experts in order to get different level of details and perspectives.

This method has medium level of time consumption meaning amount of time required for this discovery process is medium. For instance, time is spent on analysis of interviews and also time is spent on conducting interviews. Also, the interviews are iterative which makes the time spent even greater.

\*This method has high level of richness meaning insights provided about the process is high level. For instance, further clarity about issues that arise can be gained through further asking the domain experts.

\*This method has medium-high level of objectivity meaning information gathered about the process is medium-highly unbiased. For instance, the domain experts interviewed are often involved in the process and may have a conflict of interest to protect themselves and their domain. Thus, may give inaccurate or misleading responses

**Workshop-based discovery** are where information is obtained from getting key stakeholders together and letting them express their understanding of the process. Their understanding can then be challenged immediately if it conflicts with other key stakeholders understanding. This allows for quick resolving of inconsistent views between domain experts.

This method has medium level of time consumption meaning amount of time required for this discovery process is medium. For instance, workshop often requires several sessions and time is spent on conducting the workshop. This method has high level of richness meaning insights provided

about the process is high level. For instance, further clarity about issues that arise can be gained through further asking the domain experts in the workshop.

This method has medium-high level of objectivity meaning information gathered about the process is medium-highly unbiased. For instance, the domain experts interviewed are often involved in the process and may have a conflict of interest to protect themselves and their domain. Thus, may give inaccurate or misleading responses. But given they are all together it can be called out which makes the objectivity better than interview based

**Process model quality criteria** is a way of assessing whether the constructed model is of quality. Three types of qualities include-

### Semantic quality

**Semantic quality** assesses whether the model of the process matches the real-world process. The assessing is referred to as validation where the model of the process is compared with real world process in order to determine the semantic quality. This quality is assessed based on validity & completeness.

Validity describes whether all the possible paths in the model are correct and relevant to the real-world process. The domain expert is asked this.

For example- if the model indicates any junior financial officer can carry out the task of checking credit history but in real world only a senior financial officer can check. Then the model has low semantic quality since in model the junior as opposed to the senior financial officer checks credit history

For example- model is not valid since it included a path not needed

Completeness describes whether all the different instances of the process are captured

For example- model indicates when librarian arrives, she sorts the books. In real world it is described that the librarian arrives fills out entry form and then sorts book. This model has low semantic quality since in the model there is no filling out entry form task

For example- model is incomplete because it doesn't capture exceptional paths

### Syntactic quality

**Syntactic quality** assesses whether the model of the process follows the syntactic rules associated with the modelling language for our model. Important in order to ensure model can be understood and unambiguous. This quality is assessed based on structural rules & behaviour rules.

Element level rules deal with the way the different elements of the model are connected. In BPMN model the structural rules can be classified in either element level or model level. Element level rules deal with the individual elements of the model and places restrictions.

For example, in BPMN the element events have a rule where start events can't have incoming sequence flows.

Model level rules deal with ensuring all activities, gateways, sequence flows and events are on a path with a start and end events. This means the model is properly connected

For example, in model there is an isolate task with no connection to process

Behaviour rules deals with the way a model of a process can be instantiated. Must ensure the follow rules. Firstly, no activities are dead meaning all the activities must be executed in at least one instance of the process. Secondly, there needs to be proper completion meaning each instance of the process should have end event indicating completion of the instance. Thirdly, there needs to be an opportunity for the instances of the process to eventually complete

For example- in model there is no end event

### Pragmatic quality

**Pragmatic quality** assesses whether the model is clear, understandable and can be used. Difficulty in assessing the quality is that different stakeholders have different definitions of what they consider as a usable model. The standard way of accessing pragmatically quality is based on understandability, maintainability and learning

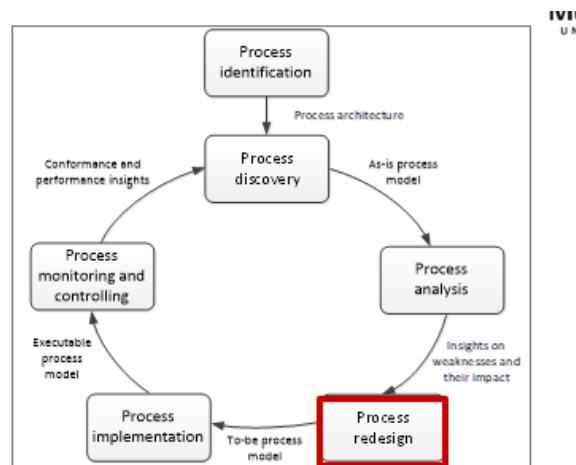
Understandability describes how simple the model is to read and comprehend.

For example- in the model there is missing labels on the flows exiting the gateway

Maintainability describes how simple it is to modify the model.

Learnability describes how good is the model is for conveying how the business process works.

## ICT393Ans7



**Process redesign** is the stage where the business process is redesigned or changed in order to resolve the issues, potential risks, and weakness identified in the process analysis stage. The AS-IS model is used as the initial starting model in which the to-be model will be generated from. Meaning the business process model is not created from scratch. The To-Be model is essentially improvements to the AS-IS model. At the start of process redesign phase, the way to assess whether a redesigned business process is an improvement or not is defined. This is basically setting the business process performance indicators. For example- a devil's quadrangle which contains cost, time, and flexibility dimensions can be used as the business process performance indicator

### Performance Indicators/Measures for success [Q1]

**Performance indicators** provide clarity as to which aspects of the process is being improved and the goals related to the improvement of the specific aspect of the process. Performance indicators provide a way to assess whether a redesigned business process is an improvement or not. The devil's quadrangle can be used as the performance indicator framework. The performance indicators in the devil's quadrangle include-

**Cost** is core measurable performance dimension for business processes. It can be viewed from different perspectives. For instance, there is fixed cost which refers to overhead cost that is not affected by the intensity of processing (rent). Another instance is variable cost which is positively related to volume of processing in business process (casual staff). Also, there is operational costs, labour cost.

NOTE: There is trade-off with cost such as decrease quality, initial investment in IT for automation (breakeven).

Type of costs:

fixed or variable

human, system (hardware/software), or external

Processing, management or support

**Time** is core measurable performance dimension for business processes. It can be viewed from different perspectives. In this context, the time is referred to as the lead time which is the time it takes to handle a transaction from start to end. A redesign aims to reduce lead time. This can mean reducing the average, maximal, or variations in lead times

It is composed of:

service time - time that resources actually spend handling the case

queue time - time spent waiting in queue because there are no resources available

waiting time - all other time

(Cycle time, throughput time)

**Quality** is can be evaluated from the clients' perspectives and workers perspective. External quality deals with client perspective and measures their satisfaction of deliverable or process. Client satisfaction with process can mean how satisfied the client is with the way the business process is carried out. Internal quality deals with workers perspective and measures satisfaction of workers

External: satisfaction of the customer

Product: product meets specification/expectation

Process: the way the product is delivered (service level)

Internal: conditions of work

Do workers feel in control?

Is work challenging?

**Flexibility** refers to the ability to react to different circumstances. Flexibility of the business process can relate to- resources, process, organisation and management

resources (ability to execute many tasks/new tasks)

process (ability to handle various cases and changing workloads)

management (ability to change rules/allocation of resources)

organisation (ability to change structure and responsiveness to meet wishes of the market and business partners)

Trade-off for automation make process less flexible

**NOTE:** Business process redesign/improvements all related to the four measurement criteria. There are trade off

### Business Process Design Best Practices [Q1]

**Business Process redesign Heuristic or Best Practices** provides a general set of possible ways to evaluate and improve an existing business process. The best practises are general meaning they may or not be applied to the specific business process. Can be classified into 6 groups but more than 30 process design best practises examples

**Task heuristics or best practices** aims to optimise individual tasks (**task elimination, task composition, task automation, ...**)

Task automation refers to tasks being performed by the system, application, tool, or technology rather than by predominately humans. Can be the implementation of automation supporting human performance of a task. Seeks to reduce human involvement in a process

Trade off less flexibility and development or investment of IT for automation is expensive

**Task elimination** - check the necessity of each task and eliminate tasks that do not add value from client's point of view. Often control tasks fall into this category (i.e. checking).

Trade-off is quality

**Routing heuristics** aims to improve flow and order of tasks in business process (**resequencing, knockout, parallelism, ...**)

Knockout is rearranging the flow of tasks, related to checking conditions that results in process termination, so that the most frequent checking condition executed is moved to the start

Trade-off is lead to longer lead time

For example- Job interviews we need degree requirement thus huge proportion is knocked out

**Parallelism** – consider whether tasks can be executed in parallel to reduce lead time. IT infrastructures which allow for the sharing of data and work enable parallelism

Trade offs include:

More complex to manage (may lead to reduction in quality and/or flexibility)

Reduces possibility of using knockout best practice

**Resource heuristics** deals with the different types of resources and amount of resources in business process (**numerical involvement, extra resources, ...**)

Empower refers to giving actors more power to handle more decisions rather than wait for middle management to always make decisions. Reduce lead time and labour cost since we remove middle management

Trade-off – quality of decisions may be reduced and errors may not be found

**Numerical involvement** – minimize the number of departments, groups and people involved in a business process

Should lead to fewer coordination problems, and hence more time for processing cases

### **Best practices for external parties**

**Integration** – consider integration of business process with a business process of client or supplier

Should be more efficient, but probably less flexible

The study by Mansar and Reijers (2007) found that BPR practitioners believed that this practice has the biggest impact on time

**Outsourcing** – consider outsourcing a business process – either wholly or partially

Intention is to reduce cost, but may reduce quality

Requires more complex coordination

**Allocation heuristics** deals with how resources are allocated in business process (**case manager, case assignment, ...**)

**Case manager** is assigned to a case. They are responsible for the case and will execute some tasks in the process

Benefits:

Having one contact person improves client satisfaction

Improves internal quality as one person is responsible for correcting mistakes

### **Integral best practices**

**Integral technology** – try to reduce the physical constraints in a business process by applying new technology to the whole process

The study by Mansar and Reijers (2007) found that BPI practitioners believed that this practice has the biggest impact on quality



## Business Process Management Suite (BPMS)

**Business Process Management Suite (BPMS) Products** refers to an application that contains a collection of technologies which supports the activities in the entire business process management lifecycle including business process identification, discovery, analysis, redesign, implementation, monitoring and controlling. Has the ability to create models of a business process be also execution of business process to test improvements

### intelligent business process management suite (iBPMS) technologies

They have more emphasis on support for system and human intelligence within business processes; e.g. simulation, optimisation and analytics

They also have added support for human collaboration, integration with social media, mobile access to processes, and real-time decision management

## ICT393Ans8

### Packaged Software Vs Building Own Software

**Buying software solutions approach** involves purchasing the required software solutions. Often considered when there is a lack of in-house personnel to maintain or develop the required software solution. However, often there a downside such as there may be a slow turnaround or additional for software modification. The software solution may not be able to achieve all the requirements or changing requirements. And may require on going prescription payments and the intellectual property may not remain in the organisation.

The advantages include-

**Cost saving** means buying the software solution is generally cheaper. Packaged software cost to develop is distributed among all the customers since the package's software is developed for a vast array of customers rather than a single customer. The support is not dedicated which makes the solution cheaper as well. On the other hand, with software built by the organisation and for the organisation it tends to be more expensive. The cost of maintaining and developing the organisation software is fully placed on the organisation. So, the organisation would need to hire many software developers or redirect work to this project.

**Time savings** means the purchased software is generally mature, exist and ready to use. Building your own software solution would mean time is lost on design, programming and testing. The packaged solution bypasses this with programming, designing and typically testing completed.

**Quality benefits** means the purchased software or packaged software generally is higher quality because it has been tested by other customers. Any faults with the existing software would have been identified by other customer and resolved. On the other hand, with software built by and for the organisation it would likely be less quality. This is because all testing and issues identification is obtained from generally only the organisation testing it. So, the amount of in-depth testing and testers is less by nature.

**Try before you buy** means that there should be readily available information about the packaged software. The packaged software can be examined based on the available information, reviews and sometimes trial use. As long as research has been done correctly the purchased product will be as expected without surprises.

**Available documentation and training** means that packaged software should have documentation about the software readily available. The useful documentation may include user guides and troubleshooting guides. The inclusion of these documentation is almost guaranteed because it serves as shield for avoiding legal liability. Lawsuits related to product use can be quickly resolved by pointing to those documents or training opportunities offered. Additionally, it enhances the software supplier's image. On the other hand, software built by the organisation and for the organisation tends to not have the documents. Documentation is normally completed last or not at all due to the common mindset that it is a time waster.

The disadvantages include-

**Lack of ownership** means the intellectual property of packaged software usually belongs to the supplier. The direction of the software in terms of its development is in the hands of supplier. Required functionality is open to be removed or modified at a whim. Also, some of the data from using the software may be transferred to the supplier or competitors.

**Failure to meet all the customer required functional requirements.** Packaged software is not designed for only one customer. So certain application specific requirements may not be provided. Client will need to accept the missing requirement or send a request for it to be developed. The supplier can reject this request. The packaged software will contain functionality not required. This is referred to as bloat and can sometimes slow the performance.

**Lack of distinctness about the software** means the packaged software can't provide the organisation with competitive edge. Competitors have access to the same or similar version of the software leaving no room for competitive difference. On the other hand, with software built by and for the organisation there is room to develop competitive difference since the organisation has large control over the development. Thus, features added to the organisation own software may not be approved in the packaged software.

Financial instability of supplier [Can do]

Difficulties meeting changing requirements

**Building software solutions approach** involves personnel in the organisation are in charge of developing the software solution the organisation requires. The advantage is if there are available resources such as personnel it may be more cost efficient also the software solution can be modified to suit the changing organisation requirements. The disadvantage is the organisation are dependent on the skills of the available personnel to develop and perhaps if the personnel leave the

organisations, it may be difficult for others to maintain the existing software solution. Also, may disadvantage the organisation in that critical resources are reallocated to develop the software.

### Software as a service (SaaS)

**Software as a service (SaaS)** is a software deliver model where software is hosted in the providers cloud server and users can access and utilise the software from different geographical approved locations. On the other hand, traditional software is run and hosted on the local machines of the organisation. Organisations will adopt SaaS over traditional software for various reasons. First, software as a service tends to be cheaper up-front meaning instead of paying for the full license the organisation will pay a fraction of the cost of the software periodically through subscription model. Therefore, the organisation has the power opt out of using the software without much cost. So as employees come and go from organisation it is more suitable for the organisation. Secondly, regular software updates tend to be free and packaged with SaaS. Traditional software required organisation to pay for the full cost of upgrading each software and required organisation to manually update it themselves. Thirdly, SaaS allows organisation existing machine infrastructure to be compatible with the software. Since it's hosted on the web any machine can access the web can utilise the software. No special machine infrastructure is needed

### Request for proposal (RFP)

**Request for proposal** is a document that an organisation creates to describe their desired software they wish to obtain for their organisation. Once the request for proposal is distributed by the organisation different proposals will sent to the organisation. The proposals are created by the potential vendors indicating their potential interest and describes the software they are willing to provide.

The request for proposal document primarily includes information about their desired software such as- functional requirements of the software, non-functional requirements of the software, the amount of money they are willing to spend to acquire the software, the date and time of when they need the software by, and context for obtaining the software. The request for proposal also includes information describing how the organisation will evaluate the proposals such as the process steps itself and evaluation criteria used. The evaluation criteria can be in the form of a weighted decision matrix which ranks which parts of the vendor proposal are most important. Also, the request for proposal describes the format of proposals they wish to receive from vendor, how vendors will submit the proposal, and organisation contract details.

### High-level Evaluation Criteria for Software

Types of high-level evaluation criteria for software –

**Ability to meet functional requirements.** This means whether the software can provide the required business functionality. This includes whether the software can support future functional requirements which are anticipated functionality that the software is predicted to be needing to support. The functional requirements are defined from taking the requirement specifications from the user and performing an analysis on it to deduce the functional requirements. Having this

criterion included provides the organisation with a way to assess whether the software provides the organisation desired software functionality. This criterion result for one software can be compared with other similar software's criterion result to work out which software is superior in providing the organisation desired functions.

**Ability to meet technical support requirements.** This refers to whether the software provider can provide the necessarily technical support the organisation desires. Having the **technical support** criterion included provides the organisation with a way to assess whether the organisation has increased risk in facing technical support issues. For example- one software provider may offer no technical support while another software provider may offer some technical support through email or calls. The software provider that does not offer any technical support may result in organisation operation being disrupted for a longer period of time if an operational issue arises. This criterion allows the selection of the software with better technical support offered.

**Ability to meet non-functional requirements.** This refers to whether the software has the required software characteristics. The software characteristics can be considered as constraints that affect how the software function are performed. Excludes the ability to provide the business functionalities. Non-functional requirement examples-

Performance requirements refers to how fast the request are responded to. So, a non-functional requirement could be the software must display a list of users in less than 3 seconds

Security requirements refers to security features of the software. For example- after 3 failed log in the account should lock

Compatibility requirement refers to hardware and operational system requirements

Having the ability to meet non-functional requirements criterion included provides the organisation with a way to assess whether the software is suitable for the context at which it's will be used in. For instance, while a software may provide all the desired functionalities if it is not compatible with our organisational infrastructure then potentially the software can't be used. For example- the software does all the functionalities required but it can't run on our computers then our employees can't use it or we would need to upgrade our computers.

**Ability to meet supplier requirements.** This is whether the supplier of the software meets the standards and expectations that the organisation expects from suppliers. The possible requirements related for suppliers include-

Financial stability refers to whether the supplier will have issues with their finance. So, if a supplier can't finically support themselves in the long run the risk of the losing the supplier in the future is high

Level of experience

Financial robustness

Having ability to meet supplier requirements criterion included is beneficial because each software suppliers have different levels of trust, reputation and reliability. Chances of potential supplier issues such as supplier going bankrupt and supplier not being able to fulfil our software modification request are reduced from utilising this criterion result as part of our consideration.

**Ability to meet initial implementation requirements.** This refers to whether the software fulfils all the requirements related to its implementation or integration into organisation. For instance, whether or not training is provided in regards to the use of the software. Also, whether the documentations for the use of the software vast, detailed and useful. And whether relevant data related to a similar old software can be converted and used in the new software.

Initial training

Hardware installation

Conversion of data

Quality of documentation

Upgrade policy

Having the ability to meet initial implementation requirements criterion included provides the organisation with a way to assess whether the organisation has increased risk in facing implementation issues. For example- one software may be able to import our old software data and another software may not. The software that does not allow for importing may result in the organisation having to create all the old data manually into the system. This criterion allows the selection of the software with smoother implementation.

**Ability to meet cost and time constraints.** This refers to whether the software cost is within the organisation preferred price range and software is available for use within a specified deadline. It helps determine whether an organisation can afford to purchase the software and when the software is needed by. Having the ability to meet cost and time constraints criterion included provides the organisation with a way to assess whether the organisation has increased risk in facing cost and time issues. For example- one software may be within our price range and another may be too expensive. This criterion allows the selection of the software that is cheaper

**Ability to meet security requirements.**

## Criteria for Software

Types of criteria for software –

Functional requirements. This means whether the software can provide the required business functionality. This includes whether the software can support future functional requirements which are anticipated functionality that the software is predicted to be needing to support. The functional requirements are defined from taking the requirement specifications from the user and performing and analysis on it to deduce the functional requirements. ~~Having this criterion included provides the organisation with a way to assess whether the software provides the organisation desired software functionality. For example one software may offer important functionality while another software may not.~~ This criterion result for one software can be compared with other similar software's criterion result to work out which software is superior in providing the organisation desired functions.

**Technical support.** This refers to whether the software provider can provide the necessarily technical support the organisation desires. ~~Having the technical support criterion included provides the organisation with a way to assess whether the organisation has increased risk in facing technical support issues. For example one software provider may offer no technical support while another~~

software provider may offer some technical support through email or calls. The software provider that does not offer any technical support may result in organisation operation being disrupted for a longer period of time if an operational issue arises. This criterion allows the selection of the software with better technical support offered.

**Non-functional requirements.** This refers to whether the software has the required software characteristics. The software characteristics can be considered as constraints that affect how the software function are performed. Excludes the ability to provide the business functionalities. Non-functional requirement examples-

Performance requirements refers to how fast the request are responded to. So, a non-functional requirement could be the software must display a list of users in less than 3 seconds

Security requirements refers to security features of the software. For example- after 3 failed log in the account should lock

Compatibility requirement refers to hardware and operational system requirements

~~Having the ability to meet non-functional requirements criterion included provides the organisation with a way to assess whether the software is suitable for the context at which it's will be used in. For instance, while a software may provide all the desired functionalities if it is not compatible with our organisational infrastructure then potentially the software can't be used. For example the software does all the functionalities required but it can't run on our computers then our employees can't use it or we would need to upgrade our computers.~~

**Supplier requirements.** This is whether the supplier of the software meets the standards and expectations that the organisation expects from suppliers. The possible requirements related for suppliers include-

Financial stability refers to whether the supplier will have issues with their finance. So, if a supplier can't finically support themselves in the long run the risk of the losing the supplier in the future is high

Level of experience

Financial robustness

~~Having ability to meet supplier requirements criterion included is beneficial because each software suppliers have different levels of trust, reputation and reliability. Chances of potential supplier issues such as supplier going bankrupt and supplier not being able to fulfil our software modification request are reduced from utilising this criterion result as part of our consideration.~~

**Initial implementation requirements.** This refers to whether the software fulfils all the requirements related to its implementation or integration into organisation. For instance, whether or not training is provided in regards to the use of the software. Also, whether the documentations for the use of the software vast, detailed and useful. And whether relevant data related to a similar old software can be converted and used in the new software.

Initial training

Hardware installation

Conversion of data

## Quality of documentation

## Upgrade policy

Having the ability to meet initial implementation requirements criterion included provides the organisation with a way to assess whether the organisation has increased risk in facing implementation issues. For example one software may be able to import our old software data and another software may not. The software that does not allow for importing may result in the organisation having to create all the old data manually into the system. This criterion allows the selection of the software with smoother implementation.

**Cost and time constraints.** This refers to whether the software cost is within the organisation preferred price range and software is available for use within a specified deadline. It helps determine whether an organisation can afford to purchase the software and when the software is needed by. Having the ability to meet cost and time constraints criterion included provides the organisation with a way to assess whether the organisation has increased risk in facing cost and time issues. For example one software may be within our price range and another may be too expensive. This criterion allows the selection of the software that is cheaper

**Ability to meet security requirements.**

## Weighted Score Approach (Criteria Not High Level)

### Weighted score method:

Criteria	Weight	Option 1		Option 2	
		Raw score	Final score	Raw score	Final score
Functional requirements	10				
Non-functional requirements	8 / 6				
Cost and time	5				
Technical support	4				
Supplier requirements	4				

### Weight score method problems include:

Assigning the weighting of importance for each criterion is hard. Often assigning the weight is subjective and consideration of other criterion weighting is required.

For example- for ... we assigned it ... and for ... we had to consider the weighting in comparison with ... Thus, working out whether ... is more important then ... is subjective

Assigning the weighting of importance for each criterion is hard especially when there are many criteria. As we weight each criterion, we are not weighting them in isolation. All criteria already weighted will need to be considered when weighing up a new criterion. Thus, as there are more criteria to consider weighting becomes more difficult and so the weight score method has problems with large number of criteria

Deceptive to compare the results of the criteria as representing an actual difference between them. This means when stakeholder read the table, they may assume ... is better than ... for our organisation. However, the definition of what is better is defined by the person who laid out the criteria. So, if the stakeholder has a different understanding of what constitutes as better then the weight score method is deceiving.

As the results are numbers, they can easily be interpreted as if they represent true differences between the alternatives – they don't (for example functional 0 all others is 10)

Difficult to define criteria so that they are independent (i.e. don't overlap) or if this can't be done, difficult to adjust weights to compensate for overlap

<https://khalilstemmler.com/articles/object-oriented/analysis/non-functional-requirements/>

## To do

- BPMN 2.0 Define Vs Activity diagram
- Draw Business process