**ICTWEB503 - Create web-based programs**

**Q1 Outline the principles of web analysis and design.**

**1. PURPOSE**

Good web design always caters to the needs of the user. Are your web visitors looking for information, entertainment, some type of interaction, or to transact with your business? Each page of your website needs to have a clear purpose, and to fulfill a specific need for your website users in the most effective way possible.

**2. COMMUNICATION**

People on the web tend to want information quickly, so it is important to communicate clearly, and make your information easy to read and digest. Some effective tactics to include in your web design include: organising information using headlines and sub headlines, using bullet points instead of long windy sentences, and cutting the waffle.

**3. TYPEFACES**

In general, Sans Serif fonts such as Arial and Verdana are easier to read online (Sans Serif fonts are contemporary looking fonts without decorative finishes).

**4. COLOURS**

A well thought out colour palette can go a long way to enhance the user experience. Complementary colours create balance and harmony. Using contrasting colours for the text and background will make reading easier on the eye. Vibrant colours create emotion and should be used sparingly. Last but not least, white space/ negative space is very effective at giving your website a modern and uncluttered look.

**5. IMAGES**

A picture can speak a thousand words, and choosing the right images for your website can help with brand positioning and connecting with your target audience. If you don’t have high quality professional photos on hand, consider purchasing stock photos to lift the look of your website. Also consider using infographics, videos and graphics as these can be much more effective at communicating than even the most well written piece of text.

**6. NAVIGATION**

Navigation is about how easy it is for people to take action and move around your website. Some tactics for effective navigation include a logical page hierarchy, using bread crumbs, designing clickable buttons, and following the ‘three click rule’ which means users will be able to find the information they are looking for within three clicks.

**7. GRID BASED LAYOUTS**

Placing content randomly on your web page can end up with a haphazard appearance that is messy. Grid based layouts arrange content into sections, columns and boxes that line up and feel balanced, which leads to a better looking website design.

**8. “F” PATTERN DESIGN**

Eye tracking studies have identified that people scan computer screens in an “F” pattern. Most of what people see is in the top and left of the screen and the right side of the screen is rarely seen. Rather than trying to force the viewer’s visual flow, effectively designed websites will work with a reader’s natural behaviour and display information in order of importance (left to right, and top to bottom).

**9. LOAD TIME**

Everybody hates a website that takes ages to load.  Tips to make page load times more effective include optimising image sizes (size and scale), combining code into a central CSS or JavaScript file (this reduces HTTP requests) and minify HTML, CSS, JavaScript (compressed to speed up their load time).

**10: MOBILE FRIENDLY**

It is now commonplace to access websites from multiple devices with multiple screen sizes, so it is important to consider if your website is mobile friendly. If your website is not mobile friendly, you can either rebuild it in a responsive layout (this means your website will adjust to different screen widths) or you can build a dedicated mobile site (a separate website optimised specifically for mobile users).

**Q2 Explain programming control structures and object-oriented programming.**

A control structure is a block of programming that analyzes variables and chooses a direction in which to go based on given parameters. The term flow control details the direction the program takes (which way program control "flows"). Hence it is the basic decision-making process in computing; flow control determines how a computer will respond when given certain conditions and parameters.

Object oriented programming is organizing a program into units where the data and code that manipulates the data are kept together, and apart from things that aren't directly relevant. That way if the data or semantics of some part change, there is an obvious place for the change to go.

**Q3 Summarise web programming concepts including:**

* + **Authentication and web security -** The process of identifying an individual, usually based on a [username](https://www.webopedia.com/TERM/U/username.html) and [password](https://www.webopedia.com/TERM/P/password.html). In [security systems](https://www.webopedia.com/TERM/S/security.html), authentication is distinct from [authorization](https://www.webopedia.com/TERM/A/authorization.html) , which is the process of giving individuals [access](https://www.webopedia.com/TERM/A/access.html) to system objects based on their [identity](https://www.webopedia.com/TERM/I/identity.html). Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual.
  + **Hypertext transfer protocol (HTTP) -** HTTPS is a protocol, which is a set of rules that governs data transfer. It is the convention how hypertext gets transferred in a network.
  + **Session management - i**s made with Session property of your context. It can be stored on your process, or on server hosted or in sql server.
  + **Stateless programming -** Stateless programming is a paradigm in which the operations functions, methods, procedures, whatever you call them you implement are not sensitive to the state of the computation. That means all the data used in an operation are passed as inputs to the operation, and all the data used by whatever operations invoked that operation are passed back as outputs

**ICTPRG604 - Create cloud computing services**

**Q4 - Identify and describe the development tools required to produce services deployable through cloud computing.**

For cloud computing to take off, there need to be tools available that enable a developer to build and deploy an application without having to download anything to their desktop. This requires an on-demand development tool that sits on top of the cloud and provides a development Platform as Service (PaaS).

There are two paths to create a development platform for cloud computing: cloud-first or tool-first.

**Cloud-first approach to PaaS**: first build a cloud platform, then build a development tool that runs on top of it.

**Tool-first approach to PaaS**: first build a development platform that is host-able tool then push that platform into the cloud.

**Q5 - Summarise the internet infrastructure necessary for cloud computing.**

Cloud infrastructure refers to the hardware and software components such as [servers](http://whatis.techtarget.com/definition/server), storage, a [network](http://searchnetworking.techtarget.com/definition/network) and [virtualization](http://searchservervirtualization.techtarget.com/definition/virtualization) software that are needed to support the computing requirements of a [cloud computing](http://searchcloudcomputing.techtarget.com/definition/cloud-computing) model.

Cloud infrastructure also includes an [abstraction layer](http://whatis.techtarget.com/definition/hardware-abstraction-layer-HAL) that virtualizes resources and logically presents them to users through [application program interfaces](http://searchexchange.techtarget.com/definition/application-program-interface) and API-enabled [command-line](http://searchwindowsserver.techtarget.com/definition/command-line-interface-CLI) or [graphical interfaces](http://searchwindevelopment.techtarget.com/definition/GUI).

In cloud computing, these virtualized resources are hosted by a service provider or IT department and are delivered to users over a network or the internet. These resources include virtual machines and components, such as servers, memory, network switches, firewalls, load balancers and storage.

**Q6 - Describe object-oriented programming and its application to cloud computing.**

Object-oriented programming (OOP) is a programming language model organized around [objects](http://searchsoa.techtarget.com/definition/object) rather than "actions" and data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data.

The ideal language for this cloud age should be object oriented because that is how we understand and are able to model the world and also embrace functional programming.

The code in cloud age is almost always distributed running on several cores/machines in the cloud centre or just the client/server separation. And it is also asynchronous. We do not block the code when waiting for WS response. The call backs come in any time.

When using standard imperative programming languages, handling the asynchrony and the distribution really complicated. You have to always take care of the current state and when the call backs come in, you have to decide what to do, in dependences of this state.

Functional programming helps to eliminate the stat" and is much better suited for this new situation. In cloud computing the code is distributed, state-less, asynchronous. Functional programming can help you with that. Object oriented is almost a must to be able to model the world.

**Q7 - Describe database access and manipulation on the web using hypertext mark-up language (HTML) extensible mark-up language (XML).**

Mark-up languages are the brick and mortar of the Web where it all started, when websites were just static pages with text and some formatting. Originating from typesetting processes used in early printing presses, these languages have long been used to annotate the text of a site, dictating both the architecture of a site and the display of text. While mark-up languages are a part of the past, that hasn’t made them obsolete. In fact, they’ve remained a core of development and its future as well.

Everything you see on the Web is a combination of mark-up (text), [CSS](https://www.upwork.com/hiring/development/css-cascading-style-sheets/) (design) and [front-end scripts](https://www.upwork.com/hiring/development/how-scripting-languages-work/) (interactivity), and that mark-up is what creates a site’s foundation. HTML is the main mark-up for web pages, or just about anything displayed in a browser, which explains why it’s still incredibly relevant, and why so many developers know it.

HTML (hypertext mark-up language) was the first Internet-based language. This descriptive mark-up language is the core mark-up for all webpages, or most anything displayed in a browser, which is why it remains such a core skill for all developers. HTML has evolved to its most recent version, [HTML5](https://www.upwork.com/hiring/development/advantages-of-html5/), which adds more features than previous versions and can now define the way videos, images, and text look.

Once a website’s HTML document is created, other aspects can be embedded into that mark-up e.g., a JavaScript program can be embedded within HTML to add interactivity to the site, and cascading style sheets (CSS) can be linked to the HTML file to control all of the design aspects. In this way, it’s the backbone of the site.

XML is another descriptive mark-up language that functions like a complement to HTML. Its core difference is that it describes elements of data (called “nodes”) while HTML displays that data. For example, in coding an online menu for a restaurant, XML tags could organize elements of each dish: the item name, description, and price. HTML tags would control how these elements were displayed: the item name in bold text, the description in italics, and the price in red.

**Describe the term "big data" and explain its relevance to cloud computing.**

**Big data**: A broad term that deals with handling of very, very large amount of data which is not possible with traditional systems. This includes collection, storage, search, analysis, visualization of the data.

Big data and cloud computing are the most hyped phenomenon driving the IT stream of businesses all around the world. Experts say that the emergence of big data was a result of the rise of cloud computing and cloud data stores. Unexceptionally, both these terminologies hold high degrees of similarity in the IT industry. They are inter-dependent in the manner that cloud resources are required to support big data projects and storage while big data is a huge business case for moving the cloud.

**ICTPRG504 - Deploy an application to a production environment**

**Q8 Describe the basic principles of database management systems**

Database is a collection of data in such a way that all the data is maintained in a structured manner. Some principles that are taken into consideration while maintaining database can be given as:

* **Data Definition:** Data definition defines the structure of the data, in which the data is maintained. By using these capabilities, a non-technical user can use the table without bothering about the formatting of the data.
* **Data dictionary:** Data dictionary means the arrangement of data of similar type together. According to the needs of the consumer, similar type of data has been stored at a single location.
* **Data manipulation language:** Data manipulation language consist of the elements that are used to access and modify the data in the table.

**Q9 Outline the software development life cycle (SDLC)**

**There are following six phases in every Software development life cycle model:**

1. **Requirement gathering and analysis -** Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements
2. **Design -** In this phase the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.
3. **Implementation or coding -** On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle.
4. **Testing -** after the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase all types of [functional testing](http://istqbexamcertification.com/what-is-functionality-testing-in-software/) like [unit testing](http://istqbexamcertification.com/what-is-unit-testing/), [integration testing](http://istqbexamcertification.com/what-is-integration-testing/), [system testing](http://istqbexamcertification.com/what-is-system-testing/), [acceptance testing](http://istqbexamcertification.com/what-is-acceptance-testing/) are done as well as [non-functional testing](http://istqbexamcertification.com/what-is-non-functional-testing-testing-of-software-product-characteristics/) are also done.
5. **Deployment -** After successful testing the product is delivered / deployed to the customer for their use.
6. **Maintenance -** Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

**Q10 Explain the coding used to create deployment applications**

Software deployment is the process of getting your program ready for market. A newly created program may work fine on your computer, but that doesn’t mean it is really ready for others to use. There are many extra program features you probably hadn’t needed for yourself, but ought to provide if the program will be used by others. These features are needed to make your program more user friendly and to help protect your program from piracy.

Consider adding the following to your program to make it ready for others to use:

* Help documents
* Executable file
* Icon library
* License Agreement
* Code Obfuscation
* Trial Version
* Install Wizard
* Uninstall program

**Q11 Explain the information and communications technology (ICT) hardware, software, security protocols and standards and organisational policies relevant to deployment of applications**

The concept of hardware includes computer components, the physical and tangible parts of the computer, i.e., electrical, electronic and mechanical parts which comprise a computer.

The term ‘software’ refers to the programs which interact with hardware to perform designated tasks. System software converts the computer, mobile phone or other device from a collection of circuits into a useful tool.

Security policy determines what your applications are allowed to do and what the users of the application are permitted to do. More importantly, they define restrictions to determine what applications and users are not allowed to do. Identify and work within the framework defined by your corporate security policy while designing your applications to make sure you do not breach policy that might prevent the application being deployed.

Security should permeate every stage of the product development life cycle and it should be a focal point of application design. Pay particular attention to the design of a solid authentication and authorization strategy. Also remember that the majority of application level attacks rely on maliciously formed input data and poor application input validation. The guidance presented in this chapter should help you with these and other challenging aspects of designing and building secure applications.

All web protocols and standards are well defined within specifications, however most web

servers and web applications accept a large number of deviations from the standard. Using

standard HTML and HTTP will help avoid the possibility of misbehavior by any of the

components interpreting the information. Whenever a non-standard protocol is used, it may

lead to ambiguity and ill-defined responses. Since it is important at each stage to

understand the expected transaction, it is best to avoid potential ambiguities.

The organization of the application plays a major role in the overall security of the site.

During the application design stage, the areas accessed without session information must

be determined. These areas may be accessed by tools that include search engines and other

non-session dependent crawlers. It is recommended that these areas be separated into

individual directories or servers to avoid mixing the public data with private areas of the

application. The simpler the application structure, the greater the chance for achieving

good security. Removing cross dependencies within the applications is an important step.

All linkages between the applications can increase the security complexity. Therefore, the

number of cross-links between the applications should be limited and clearly mapped with

their security considerations understood.

**ICTICT511 - Match ICT needs with the strategic direction of the enterprise**

**Q12 Identify methods of analysis and planning approaches to technical problems and management requirements**

**1. Define the problem**

This is often where people struggle. They react to what they think the problem is. Instead, seek to understand more about why you think there's a problem**.**

**2. Look at potential causes for the problem**

It's amazing how much you don't know about what you don't know. Therefore, in this phase, it's critical to get input from other people who notice the problem and who are effected by it.

**3. Identify alternatives for approaches to resolve the problem**

A wonderful set of skills used to identify the underlying cause of issues is Systems Thinking.

**4. Select an approach to resolve the problem**

* When selecting the best approach, consider:
* Which approach is the most likely to solve the problem for the long term?
* Which approach is the most realistic to accomplish for now? Do you have the resources? Are they affordable? Do you have enough time to implement the approach?
* What is the extent of risk associated with each alternative?

**5. Plan the implementation of the best alternative (this is your action plan)**

* Carefully consider "What will the situation look like when the problem is solved?"
* What steps should be taken to implement the best alternative to solving the problem? What systems or processes should be changed in your organization, for example, a new policy or procedure? Don't resort to solutions where someone is "just going to try harder".
* How will you know if the steps are being followed or not? (these are your indicators of the success of your plan)
* What resources will you need in terms of people, money and facilities?
* How much time will you need to implement the solution? Write a schedule that includes the start and stop times, and when you expect to see certain indicators of success.
* Who will primarily be responsible for ensuring implementation of the plan?
* Write down the answers to the above questions and consider this as your action plan.
* Communicate the plan to those who will involved in implementing it and, at least, to your immediate supervisor.

**6. Monitor implementation of the plan**

Monitor the indicators of success:

* Are you seeing what you would expect from the indicators?
* Will the plan be done according to schedule?
* If the plan is not being followed as expected, then consider: Was the plan realistic? Are there sufficient resources to accomplish the plan on schedule? Should more priority be placed on various aspects of the plan? Should the plan be changed?

**7. Verify if the problem has been resolved or not**

One of the best ways to verify if a problem has been solved or not is to resume normal operations in the organization.

**Q13 Assess and document the hardware platform used by the organisation and network, and the subsequent security guidelines required**

Most of the Organisation and network are using an intel or amd hardware platform as these is the most popular hardware platform that you can use as of today.

Both of these platforms are using almost the same security guidelines. Common guidelines that they use:

**Transaction security**

• Always translate incoming requests into a standard encoding scheme.

• Verify the maximum number of characters for all fields.

• Verify that the input does not contain dangerous characters.

• Avoid free format input, put as many constraints on it as possible.

• Never use values received from the client to directly create dynamic pages.

• Obscure your transaction by using POST method instead of GET.

• Parameters passed through the client must be encrypted and signed.

• Remove all meta-information from the information sent to the client.

• Use only standard protocols in the application.

**Session Security**

• Authentication information must always be encrypted.

• Strong password schemes must be used to prevent enumeration.

• Require secondary authentication for critical parts of the application.

• Never leave default accounts on the application.

• Use defensive measures to counter authentication attacks.

• All private areas of the application should be associated with a session identifier.

• The client or client code should never change session identifiers.

• Session must be terminated under the following conditions: inactive session, long

session, logoff and security error.

• Always define all the possible states of the application during design stage.

• All web pages should belong to one of the states.

**Application Security**

• Areas without any session information must be identified during design stage.

• Application entry points must be established during design stage,

• Minimize the number of cross-dependencies between applications.

• Encrypt the stream all private areas of the application.

• Do not make links dependent on the encryption to allow use of proxy.

• Encrypt and sign critical information (in addition to the stream encryption).

• Always use the “no-cache” indicator for private information.

•

**Application Environment**

• Never develop on the production server- always copy to it from an internal server.

• Keep only the minimal part of the application on the production server.

• Never put any content pages on external cache server.

• Separate your application to make use of the DMZ.

**3rd party tools**

• Use the vendors’ guidelines for maximum-security installation.

• Remove all default accounts and change any default passwords.

• Remove all demo and sample parts of the application.

• Follow an updated list of patches for all the tools.

• Minimize the length of the patch latency.

**Q14 Explain current system functionality to forecast for planning**

**• Understand the work –** Relate capacity requirements to what work needs to be supported

**• Translate –** Translate the capacity requirements necessary to support business functions into

technical requirements

**• Evaluate –** Constantly evaluate current capacity against requirements defined by the client

**• Plan** – Plan for future capacity requirements and the incremental step necessary to achieve them

**• Research –** Research other projects, their capacities, and utilizations. If resources are not being fully

utilized explore the possibility of leveraging other projects to reduce cost and time to delivery

**• Business Requirements –** Confirm that the individuals measuring and estimating capacity

requirements understand the business requirements driving the needs for the capacity

**• Change Control –** A key management component is the practice of managing change

**• Tools –** Some tools for managing capacity may include:

**o Virtualization Software** – manage computers, share resources, and save money

**o Source Control Management Tools** – improve software development capacity

**o Sub-Contractors –** manage temporary fluctuations in staffing requirements

**o Virtualization Software –** manage computers, share resources, and save money

**o Source Control Management Tools –** improve software development capacity

**o Sub-Contractors –** manage temporary fluctuations in staffing requirements

**• Types of Capacity –** The following list is in no way a full range of capacity items. When planning

capacity requirements consider items such as physical server consolidation, storage, necessary IT

controls, standards and processes, computing resources, required technical skills, additional staffing,

requirements for licensing, backups, security, system configuration, monitoring, and availability, etc.

**Q15 Identify and specify operating systems supported by the organisation to forecast for planning**

Forecasting is the process of making predictions of the future based on past and present data and most commonly by analysis of trends. A commonplace example might be estimation of some variable of interest at some specified future date. Prediction is a similar, but more general term. Both might refer to formal statistical methods employing time series, cross-sectional or longitudinal data, or alternatively to less formal judgmental methods. Usage can differ between areas of application: for example, in hydrology the terms "forecast" and "forecasting" are sometimes reserved for estimates of values at certain specific future times, while the term "prediction" is used for more general estimates, such as the number of times floods will occur over a long period.

Risk and uncertainty are central to forecasting and prediction; it is generally considered good practice to indicate the degree of uncertainty attaching to forecasts. In any case, the data must be up to date in order for the forecast to be as accurate as possible.

Organizations can select any operating system they want to use for forecast planning.

Forecasting is a common statistical task in ICT, where it helps to inform decisions about the scheduling of production, personnel and etc. It provides a guide to long-term strategic planning. However forecasting is often done poorly, and is frequently confused with planning and goals.

**Q16Compare competing and complementary internal and external operating environments**

Following on from the definition of a Standard, ICT Services actively monitors and evaluates changes in a range of ICT products (hardware and software) and associated technologies. This work is overseen by the ICT Services Committee in conjunction with the ICT Information Standards Committee. It is with the application of these concepts in mind that ICT has introduced the common or Standard Operating Environment (SOE). We recognise that some customers have specialised IT requirements and we are able to customise our services to suit these. However, we also recognise that the majority of ICT customers only require a common configuration on their computer workstation. By standardising on the SOE, we endeavor.

A Standard Operating Environment (SOE) is a standard implementation of an operating system and its associated software. Associated names and concepts include:

MOE - Managed Operating Environment

COE - Consistent or Common Operating Environment

MDE - Managed Desktop Environment

DMS - Desktop Managed Services

OSP – On screen programming

SDE - Standard Desktop Environment

SDC - Standard Desktop Configuration

UOE - Unmanaged Operating Environment

For any business to grow and prosper, managers of the business must be able to anticipate, recognise and deal with change in the internal and external environment. Change is a certainty, and for this reason business managers must actively engage in a process that identifies change and modifies business activity to take best advantage of change. That process is strategic planning.

All organization have an internal and external environment. The internal environment is very much associated with the human resource of the business or organisation, and the manner in which people undertake work in accordance with the mission of the organisation. To some extent, the internal environment is controllable and changeable through planning and management processes.

The external environment, on the other hand is not controllable. The managers of a business have no control over business competitors, or changes to law, or general economic conditions. However the managers of a business or organisation do have some measure of control as to how the business reacts to changes in its external environment.

**Q17 Identify and assess technology and product directions for evaluating and forecasting vendor and technology trends.**

A major problem for firms making information technology investment decisions is predicting and understanding the effects of future technological developments on the value of present technologies. Failure to adequately address this problem can result in wasted organization resources in acquiring, developing, managing, and training employees in the use of technologies that are short-lived and fail to produce adequate return on investment. The sheer number of available technologies and the complex set of relationships among them make IT landscape analysis extremely challenging. Most IT-consuming firms rely on third parties and suppliers for strategic recommendations on IT investments, which can lead to biased and generic advice. We address this problem by defining a new set of constructs and methodologies upon which we develop an IT ecosystem model. The objective of these artifacts is to provide a formal problem representation structure for the analysis of information technology development trends and to reduce the complexity of the IT landscape for practitioners making IT investment decisions. We adopt a process theory perspective and use a combination of visual mapping and quantification strategies to develop our artifacts and a state diagram-based technique to represent evolutionary transitions over time. We illustrate our approach using two exemplars: digital music technologies and wireless networking technologies.

**ICTPRG520 - Validate an application design against specifications**

**Q18 Explain the database design and implementation**

Databases are at the centre of most information systems in everyday use, therefore it is important that they are designed and built using appropriate methods to ensure that they meet users’ requirements whilst being robust and maintainable. A database system is usually regarded as the database which contains related tables of data maintained by a database management system (DBMS), along with applications that provide controlled access to the database. In order to build an effective database system it is important to understand and apply the database development lifecycle, which includes the following phases:

**1. Strategy and planning –** typically the cycle starts with the strategy and planning phase to identify the need and scope of a new system.

**2. Requirements analysis phase –** a more detailed requirements analysis will be carried out which will include identifying what the users require of the system; this will involve conceptual analysis.

**3. Design phase –** this will involve producing a conceptual, logical and physical design. To undertake these processes it is important to be able to understand and apply the data modelling techniques which are covered in this book. When a suitable logical design has been obtained the development phase can begin.

**4. Development phase** – this involves creating the database structure using an appropriate Database

Management System (DBMS) and usually includes the development of applications that provide a

user interface consisting of forms and reports which will allow controlled access to the data held in

the database. This book will show how the Oracle relational database management system and the

Oracle Application Express (APEX) application developer tool can be used for this purpose.

**5. Deployment/implementation** – when the system has been developed it will be tested, it will then be

deployed ready for use.

**6. Operations and maintenance** – following the system release for use it will be maintained until it reaches the end of its useful life, at this stage the development lifecycle may restart.

**Q19 Describe business and technical modelling using UML tools at intermediate level**

Business process modeling is mainly used to map a workflow so you can understand, analyse and make positive changes to that workflow or process.

UML is a modeling language mainly used for specification, visualization, development and documenting of software systems. But business professionals have adapted it as a powerful business process modeling technique.

With 14 different UML diagram types, it offers a flexible and powerful way to visualize almost any business process. They are typically used for modeling the detailed logic of a business process. In many ways, UML diagrams are the object-oriented equivalent of flow charts.

As mentioned above one of its main advantages is its flexibility. But with 14 different diagram types, some might find it difficult to understand the diagrams. Add to that the same process can be modeled using different UML diagrams. So probably not the most popular choice among analysts.