

INDUSTRIAL VISIT REPORT

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF SCIENCE

(COMPUTER SCIENCE)

SUBMITTED BY

MUHAMMED ABDUL BASITH(RCAVSCS006)

UNDER THE GUIDANCE OF

Mr. ABDUL LATHEEF A P

(Asst. Prof. of the Department of Computer Science)

SOFTWARE DEVELOPMENT STRATEGIES AND TECHNOLOGIES



Regional College of Science and Humanities

Kizhisseri

2021-2024



DEPARTMENT OF B.Sc COMPUTER SCIENCE

Kizhisseri, Puliyakode (P.O) Malappuram (Dist)

CERTIFICATE

This is to certify that Mr. **MUHAMMED ABDUL BASITH (RCAVSCS006)**, is the student of the department of B.Sc Computer Science, **REGIONAL COLLEGE OF SCIENCE AND HUMANITIES** and the Industrial Visit report has been prepared by him in partial fulfilment of the requirement for the award of B.Sc Computer Science degree of the University of Calicut.

Place: Kizhisseri

Date: 22/03/2024

Head of the Department

Mr. ABDUL LATHEEF A P



DEPARTMENT OF B.Sc COMPUTER SCIENCE

Kizhisseri, Puliyakode (P.O) Malappuram (Dist)

CERTIFICATE

This is Certify that this Industrial Visit Report is submitted to Calicut University in partial fulfilment of the requirement of the award of the degree B.Sc Computer Science report of the original work done by Mr. **MUHAMMED ABDUL BASITH (Reg No RCAVSCS006)**, during the year 2021-2024 in the department of computer science, **REGIONAL COLLEGE OF SCIENCE AND HUMANTIES.**

Date: 22/03/2024

Head of the Department

Submitted for the university project Viva-Voice examination held on

Internal Examiner

External Examiner

DECLARATION

I, **MUHAMMED ABDUL BASITH (RCAVSCS006)** student of **REGIONAL COLLEGE OF SCIENCE AND HUMANITIES, KIZHISSERI** have completed the INDUSTRIAL VISIT for the academic year **2021-2024**. The information given in this report is true to the best of my knowledge.

PLACE: Kizhisseri

NAME: MUHAMMED ABDUL BASITH

DATE: 22/03/2024

CERTIFICATE OF TRAINING

Date: 18.03.2024

TO WHOM IT MAY CONCERN

This is to certify that **MUHAMMED ABDUL BASITH (Reg. No. RCAVSCS006)** student of **REGIONAL COLLEGE OF SCIENCE AND HUMANITIES** has successfully completed his academic main project entitled “**CLUB MANAGEMENT SYSTEM**” in **FLUTTER AND PYTHON - DJANGO** under the guidance of our senior developers during the period December 2023 to March 2024.

During this period, he was found hardworking, punctual and efficient. We wish him a successful future.

Project Manager



NIKHIL P



INDEX

No	Training
1	About Organization
2	Staff Hierarchy and roles
3	Services
4	Development Strategies
5	Technologies
6	Training Based on Development Strategies
7	Product Details



ORGANIZATION OVERVIEW

NEXUS TECHNOLOGIES is one of the pioneers in Indian IT sector, with its origins in 2013. Over this duration we have developed and implemented solutions for multiple market segments, across the range of technologies in India. We have been in the forefront in introducing new technologies and solutions.

NEXUS TECHNOLOGIES is India's premier Embedded System, Software Development and Training company offering a wide range of technologies including IoT, Robotics and web development. Also, our training wing is providing various courses and internship training programs for academic students to fulfil their dream in IT career.

MISSION AND VISION

The mission is to provide comprehensive solution to all technical problems concerning the software solution, competitions, learning, development activities and generation of interest in science and technologies among student and to become a name synonymous with workshop and training programs by providing the most comprehensive and fulfilling services.

Our dream is to change the overall education system of India and to develop the students with the upcoming technologies, so that they can compete with the world, by providing them latest technology and updated workshop and training

SERVICES

- WEB DEVELOPMENT
- EMBEDDED SYSTEM DEVELOPMENT
- PCB DESIGNING AND FABRICATION
- ROBOTICS DEVELOPMENT
- INTERNET OF THINGS (IoT)

TECHNOLOGY OVERVIEW

- PYTHON
- ANDROID
- EMBEDDED C
- JAVA

SERVICES FOR EDUCATION SYSTEM

- Technical Workshops
- Internship Training Programs
- Technology Courses
- Specialized in setting up labs for educational institutions

Chief Executive Officer

o Chief Marketing Officer

- Marketing Managers
- Business Development Executives

o Chief Operating Officer

- Sr. Project Manager
- Project Managers

o Team Leaders

- Sr. Software Engineers
- Software Engineers

o Associate Engineers

Services

Software Development

NEXUS Technologies develops an optimal enterprise software solution with regard to your business needs and costs/productivity statement using our best practices in software development. Our technological expertise includes years of working with .NET, PHP, JavaScript and other technologies. We offer a network of client-based solutions in automation and re-engineering your manual operations through standardized processes. For the improvement of efficiency in a cost-effective manner the initial system study and high-level design are carried out on-site by domain experts, followed by detailed design, development that are carried out by our specialists. We employ unique strategies and business practices to deliver unparalleled MIS solutions. We always look forward for advancement and helps empower enterprises with business process innovation, enterprise agility, collaboration across the ecosystem and utilization of information as a strategic asset. Our solutions have provided valuable insights and help transform the business processes of clients. From initial concept to final implementation, we carefully assess your competitive needs to deliver a solution that has a positive impact on your overall organization. We have a proactive approach, which results in minimal disruption to your business. We take care of your application and constant design enhancement for which we use our exclusive tools and methodologies to quickly resolve the problem.

Web applications

We provide custom web application development services including web site design and development. It also includes software consulting and implementing outstanding e-commerce and CMS solutions. We help clients in choosing the right technology thus ensuring that the ultimate decision is driven completely by client's preferences and IT requirements. On each development process we consider the newness of the Internet and World Wide Web. We always keep us informed about the latest developments in technology and we make sure that our clients get the maximum use of it depending up on their needs. We also specialize in web development of dynamic database driven web sites. Our web development projects are partial or full CMS with custom written admin panels tailored to specific client's need. Our web design services give you the power to communicate effectively with both prospects and customers. We handle all aspects

of web development project from client's initial to following consultations, meetings or discussions on the phone as needed. We help clients in choosing the technology ensuring that the ultimate decision is driven completely by client's preferences and IT requirements. On each development process we consider the newness of the Internet and World Wide Web. We always keep us informed about the latest developments in technology and we make sure that our clients get the maximum use of it depending up on their needs. We also specialize in web development of dynamic database driven web sites. Our web development projects are partial or full CMS with custom written admin panels tailored to specific client's need. Our web design services give you the power to communicate effectively with both prospects and customers. We handle all aspects of web development project from client's initial to following consultations, meetings or discussions on the phone as needed.

Android Applications

With an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Android market is developing fast as the operating system is giving a lot of user-friendly applications. Google has radically changed the smart phone industry with its android operating system. Before being into this we need to have a solid plan. With NEXUS TECHNOLOGIES you can chart out a mobile strategy and implement it effectively.

ANDROID SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android. If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in ADT (Android Developer Tools) to streamline your Android app development.

What is Android?

Android is the world's most popular operating system for mobile devices and tablets. It is an open-source operating system, created by Google, and available to all kinds of developers with various expertise levels, ranging from rookie to professional.

From a developer's perspective, Android is a Linux-based operating system for smartphones and tablets. It includes a touch screen user interface, widgets, camera, network data monitoring and all the other features that enable a cell phone to be called a smartphone. Android is a platform that supports various applications, available through the Android Play Store. The Android platform also allows end users to develop, install and use their own applications on top of the Android framework. The Android framework is licensed under the Apache License, with Android application developers holding the right to distribute their applications under their customized license. Android is a powerful Operating System supporting a large number of applications in Smart Phones

Android comes with an Android market which is an online software store. It was developed by Google. It allows Android users to select, and download applications developed by third party developers and use them. There are around 2.0 lack+ games, application and widgets available on the market for users.

Android applications are written in java programming language. Android is available as open source for developers to develop applications which can be further used for selling in android market. There are around 200000 applications developed for android with over 3 billion+ downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. For software development, Android provides **Android SDK** (Software development kit).

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit, but other development tools are available.

A software development kit (SDK or "devkit") is typically a set of software development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.

It may be something as simple as the implementation of one or more application programming interfaces (APIs) in the form of some libraries to interface to a particular programming language or to include sophisticated hardware that can communicate with a particular embedded system. Common tools include debugging facilities and other utilities, often presented in an integrated development environment (IDE). SDKs also frequently include sample code and supporting technical notes or other supporting documentation to help clarify points made by the primary reference material.

SDKs may have attached licenses that make them unsuitable for building software intended to be developed under an incompatible license. For example, a proprietary SDK will probably be incompatible with free software development, while a GPL-licensed SDK could be incompatible with proprietary software development. LGPL SDKs are typically safe for proprietary development.

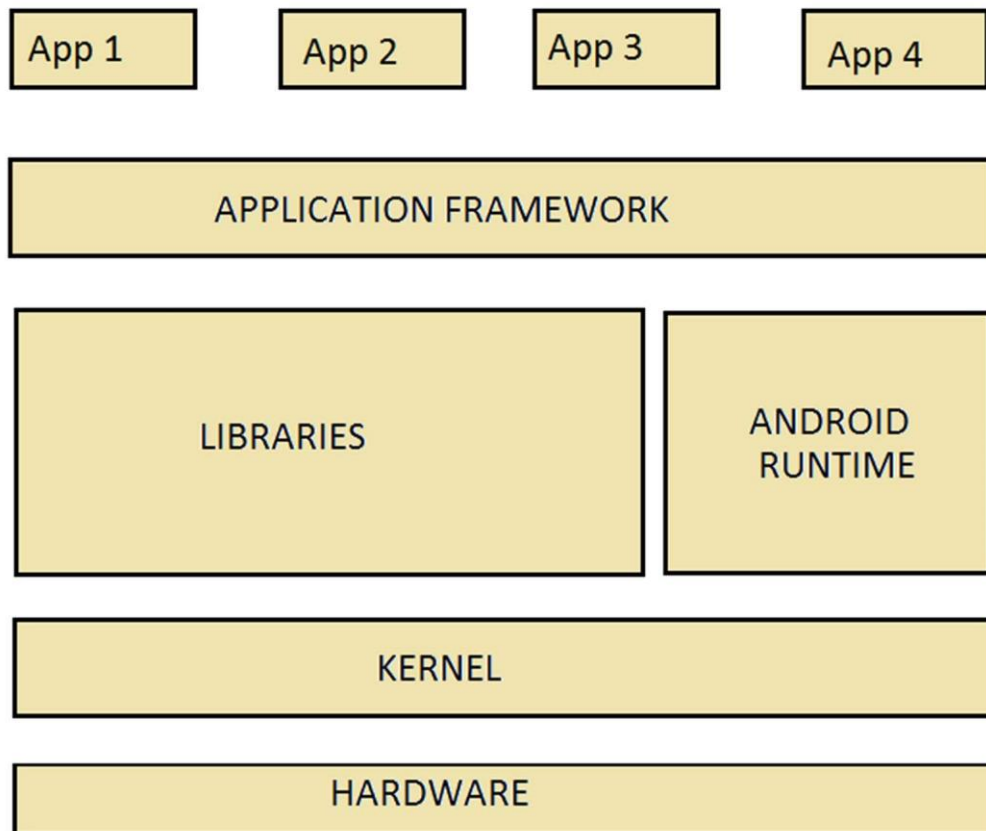
A software engineer typically receives the SDK from the target system developer. Often the SDK can be downloaded directly via the Internet. Many SDKs are provided for free to encourage developers to use the system or language. Sometimes this is used as a marketing tool.

An SDK for an operating system add-on (for instance, QuickTime for Mac OS) may include the add-on software itself, to be used for development purposes if not necessarily for redistribution together with the developed product. An interesting situation arises here between platforms where it is possible to develop applications that can at least start up on a system configuration without the add-on installed, and use a Gestalt-style run-time environment query to determine if the add-on is present, and ones where the application will simply fail to start. In other words, it is possible to build a single binary that will run on configurations with and without the add-on present, albeit operating with reduced functionality in the latter situation.

Understanding Android

To begin development on Android even at the application level, I think it is paramount to understand the basic internal architecture. Knowing how things are arranged inside helps us understand the application framework better, so we can design the application in a better way.

Android is an OS based on Linux. Hence, deep inside, Android is pretty similar to Linux. To begin our dive into the Android internals, let us look at an architectural diagram.



Applications

The diagram shows four basic apps (App 1, App 2, App 3 and App 4), just to give the idea that there can be multiple apps sitting on top of Android. These apps are like any user interface you use on Android; for example, when you use a music player, the GUI on which there are buttons to play, pause, seek, etc is an application. Similarly, is an app for making calls, a camera app, and so on. All these apps are not necessarily from Google. Anyone can develop an app and make it

available to everyone through Google Play Store. These apps are developed in Java, and are installed directly, without the need to integrate with Android OS.

Application Framework

Scratching further below the applications, we reach the application framework, which application developers can leverage in developing Android applications. The framework offers a huge set of APIs used by developers for various standard purposes, so that they don't have to code every basic task. The framework consists of certain entities; major ones are:

- **Activity Manager**

This manages the activities that govern the application life cycle and has several states. An application may have multiple activities, which have their own life cycles. However, there is one main activity that starts when the application is launched. Generally, each activity in an application is given a window that has its own layout and user interface. An activity is stopped when another starts, and gets back to the window that initiated it through an activity call back.

- **Notification Manager**

This manager enables the applications to create customized alerts

- **Views**

Views are used to create layouts, including components such as grids, lists, buttons, etc.

- **Resource Managers**

Applications do require external resources, such as graphics, external strings, etc. All these resources are managed by the resource manager, which makes them available in a standardized way.

- **Content Provider**

Applications also share data. From time to time, one application may need some data from another application. For example, an international calling application will need to access the user's address book. This access to another application's data is enabled by the content providers.

Libraries

This layer holds the Android native libraries. These libraries are written in C/C++ and offer capabilities similar to the above layer, while sitting on top of the kernel. A few of the major native libraries include

- Surface Manager: Manages the display and compositing window-ing manager. - Media framework: Supports various audio and video formats and codecs including their playback and recording.
- System C Libraries: Standard C library like libc targeted for ARM or embedded devices.
- OpenGL ES Libraries: These are the graphics libraries for rendering 2D and 3D graphics.
- SQLite: A database engine for Android.

Android Runtime

The Android runtime consists of the Dalvik Virtual Machine. It is basically a virtual machine for embedded devices, which like any other virtual machine is a bytecode interpreter. When we say it is for embedded devices, it means it is low on memory, comparatively slower and runs on battery power. Besides the Dalvik Virtual Machine, it also consists of the core libraries, which are Java libraries and are available for all devices.

Kernel

The Android OS is derived from Linux Kernel 2.6 and is actually created from Linux source, compiled for mobile devices. The memory management, process management etc. are mostly similar. The kernel acts as a Hardware Abstraction Layer between hardware and the Android software stack.

JAVA

Initially the language was called as “oak” but it was renamed as “java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

Java is a programmer’s language

- ☐ Java is cohesive and consistent
- ☐ Except for those constraint imposed by the Internet environment. Java gives the programmer, full control

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs in the areas of security and probability. But Java addresses these concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

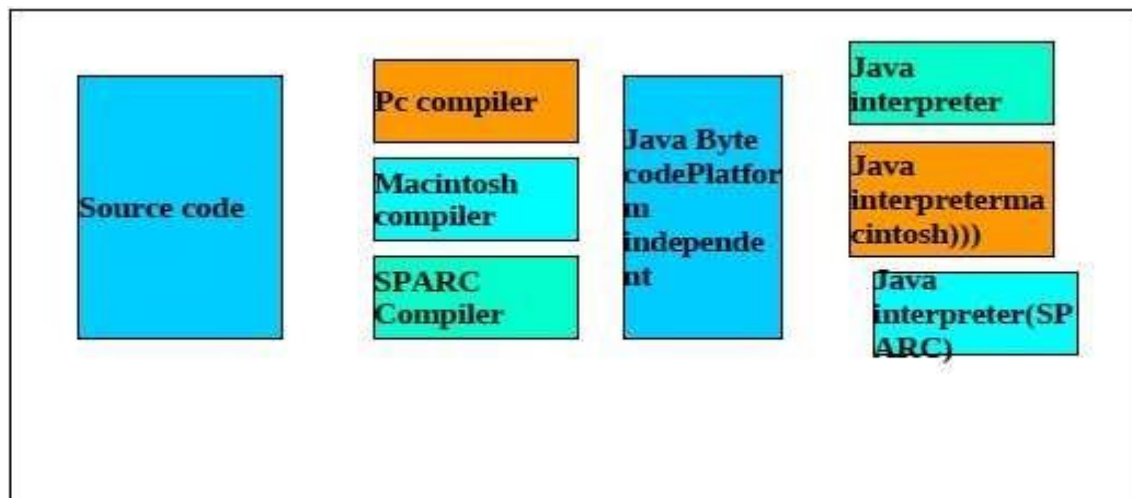
Compilation of code

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine(JVM). The JVM is supposed to execute the byte code. The JVM is created for the overcoming the issue of probability. The code is written and

compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

Compiling and interpreting java source code.

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets.



Features of java

1)**Compiled and Interpreter:** - has both Compiled and Interpreter Feature Program of java is First Compiled and then it is must to Interpret it. First of all, The Program of java is Compiled then after Compilation it creates Bytes Codes rather than Machine Language Then After Bytes Codes are Converted into the Machine Language is Converted into the Machine Language with the help of the Interpreter So for Executing the java Program First of all it is necessary to Compile it then it must be Interpreter

2)**Platform Independent:** - Java Language is Platform Independent means program of java is Easily transferable because after Compilation of java program bytes code will be created then we have to just transfer the Code of Byte Code to another computer. This is not necessary for computer s having same Operating System in which the code of the java is Created and Executed

After Compilation of the Java Program We Easily Convert the Program of the java top another Computer for Execution

3)**Object-Oriented:** - We Know that is purely OOP Language that is all the Code of the java Language is Written into the classes and Objects So for This feature java is Most Popular Language because it also Supports Code Reusability, Maintainability etc.

4)**Robust and Secure:** - The Code of java is Robust and Mean sot first checks the reliability of the code before Execution When We trying to Convert the Higher data type into the Lower Than it Checks the Demotion of the Code the It Will Warns a User to Not to do this So it is called as Robust

5)**Secure:** When We convert the Code from One Machine to Another the First Check the Code either it is Affected by the Virus or not or it Checks the Safety of the Code if code contains the Virus then it will never Executed that code on to the Machine

6) **Distributed:** - Java is Distributed Language Means because the program of java is compiled onto one machine can be easily transferred to machine and executes them on another machine because facility of Bytes Codes So java is Specially designed For Internet Users which uses the Remote Computers for Executing their Programs on local machine after transferring the Programs from Remote Computers or either from the internet.

7)**Simple Small and Familiar:** - is a simple Language Because it contains many features of other Languages like c and C++ and Java Removes Complexity because it doesn't use pointers, Storage Classes and Go to Statements and java Doesn't support Multiple Inheritance

8)**Multithreaded and Interactive:-** Java uses Multithreaded Techniques For Execution Means Like in other in Structure Languages Code is Divided into the Small Parts Like These Code of java is divided into the Smaller parts those are Executed by java in Sequence and Timing Manner this is Called as Multithreaded In this Program of java is divided into the Small parts those are Executed by Compiler of java itself Java is Called as Interactive because Code of java Supports Also CUI and Also GUI Programs

9) **Dynamic and Extensible Code:** - Java has Dynamic and Extensible Code Means With the Help of OOPS java Provides Inheritance and With the Help of Inheritance we Reuse the Code that is Pre-defined and also uses all the built in Functions of java and Classes

10) **Distributed:** - Java is a distributed language which means that the program can be design to run on computer networks. Java provides an extensive library of classes for communicating, using TCP/IP protocols such as HTTP and FTP. This makes creating network connections much easier than in C/C++. You can read and write objects on the remote sites via URL with the same ease that programmers are used to when read and write data from and to a file. This helps the programmers at remote locations to work together on the same project.

11) **Secure:** Java was designed with security in mind. As Java is intended to be used in networked/distributor environments so it implements several security mechanisms to protect you against malicious code that might try to invade your file system.

For example: The absence of pointers in Java makes it impossible for applications to gain access to memory locations without proper authorization as memory allocation and referencing model is completely opaque to the programmer and controlled entirely by the underlying run-time platform.

12) **Architectural Neutral:** One of the key features of Java that makes it different from other programming languages is architectural neutral (or platform independent). This means that the programs written on one platform can run on any other platform without having to rewrite or recompile them. In other words, it follows 'Write-once-run-anywhere' approach. Java programs are compiled into byte-code format which does not depend on any machine architecture but can be easily translated into a specific machine by a Java Virtual Machine (JVM) for that machine. This is a significant advantage when developing applets or applications that are downloaded from the Internet and are needed to run on different systems.

13) **Portable:** The portability actually comes from architecture-neutrality. In C/C++, source code may run slightly differently on different hardware platforms because of how these platforms implement arithmetic operations. In Java, it has been simplified.

Unlike C/C++, in Java the size of the primitive data types are machine independent. For example, an int in Java is always a 32-bit integer, and float is always a 32-bit IEEE 754 floating point number. These consistencies make Java programs portable among different platforms such as Windows, UNIX and Mac.

14) **Interpreted:** Unlike most of the programming languages which are either compiled or interpreted, Java is both compiled and interpreted. The Java compiler translates a java source file to byte codes and the Java interpreter executes the translated byte codes directly on the system that implements the Java Virtual Machine. These two steps of compilation and interpretation allow extensive code checking and improved security.

15) **High performance:** Java programs are compiled to portable intermediate form known as byte codes, rather than to native machine level instructions and JVM executes Java byte code on any machine on which it is installed. This architecture means that Java programs are faster than program or scripts written in purely interpreted languages but slower than C and C++ programs that compiled to native machine languages.

Although in the early releases of Java, the interpretation of by byte code resulted in slow performance but the advance version of JVM uses the adaptive and just in time (JIT) compilation technique that improves performance by converting Java byte codes to native machine instructions on the fly.

Java Server Pages (JSP):

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP and ASP, but it uses the Java programming language. To deploy and run Java Server Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required.

Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime, each JSP servlet is cached and re-used until the original JSP is modified. JSP can be used independently or as the view component of a server-side model–view–

controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture.

JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, such as HTML, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other Java program, they must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of OutputStream, they can deliver other types of data as well. The Web container creates JSP implicit objects like request, response, session, application, config, page, pageContext, out and exception. JSP Engine creates these objects during translation phase.

JSP pages use several delimiters for scripting functions. The most basic is `<% ... %>`, which encloses a JSP scriptlet. A scriptlet is a fragment of Java code that is run when the user requests the page. Other common delimiters include `<%= ... %>` for expressions, where the scriptlet and delimiters are replaced with the result of evaluating the expression, and directives, denoted with `<%@ ... %>`. Java code is not required to be complete or self-contained within a single scriptlet block. It can straddle markup content, provided that the page as a whole is syntactically correct. For example, any Java if/for/while blocks opened in one scriptlet must be correctly closed in a later scriptlet for the page to successfully compile.

Content which falls inside a split block of Java code (spanning multiple scriptlets) is subject to that code. Content inside an if block will only appear in the output when the if condition evaluates to true. Likewise, content inside a loop construct may appear multiple times in the output, depending upon how many times the loop body runs.

Development Strategies

WATERFALL MODEL

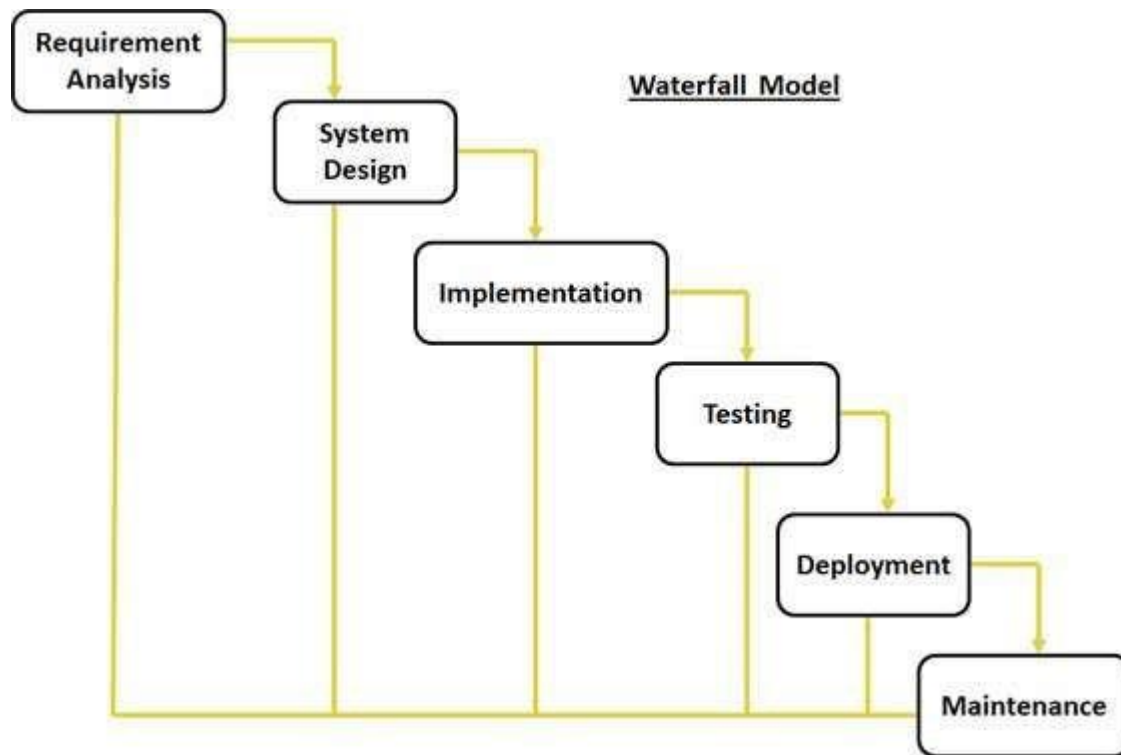
The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Waterfall model is the earliest SDLC approach that was used for software development .

The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap.

Waterfall Model design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Following is a diagrammatic representation of different phases of waterfall model.



The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.

- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model Pros & Cons

ADVANTAGE

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

DISADVANTAGE

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The following table lists out the pros and cons of Waterfall model:

Pros	Cons
Simple and easy to understand and use	No working software is produced until late during the life cycle.
Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.	High amounts of risk and uncertainty.
Phases are processed and completed one at a time.	Not a good model for complex and object-oriented projects.
Works well for smaller projects where requirements are very well understood.	Poor model for long and ongoing projects.
Clearly defined stages.	Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model.
Well, understood milestones.	It is difficult to measure progress within stages.
Easy to arrange tasks.	Cannot accommodate changing requirements.
Process and documented. results are well documented	No working software is produced until late in the life cycle.
	Adjusting scope during the life cycle can end a project.
	Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

AGILE DEVELOPMENT METHOD

Agile development provides opportunities to assess the direction throughout the development lifecycle. This is achieved through regular cadences of work, known as Sprints or iterations, at the end of which teams must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental.” In waterfall, development teams only have one chance to get each aspect of a project right. In an agile paradigm, every aspect of development — requirements, design, etc. — is continually revisited. When a team stops and reevaluates the direction of a project every two weeks, there’s time to steer it in another direction.

This “inspect-and-adapt” approach to development greatly reduces development costs and time to market. Because teams can develop software at the same time they’re gathering requirements, “analysis paralysis” is less likely to impede a team from making progress. And because a team’s work cycle is limited to two weeks, stakeholders have recurring opportunities to calibrate releases for success in the real world. Agile development helps companies build the right product. Instead of committing to market a piece of software that hasn’t been written yet, agile empowers teams to continuously replan their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. Agile development preserves a product’s critical market relevance and ensures a team’s work doesn’t wind up on a shelf, never released.

What Is Agile?

The Agile movement proposes alternatives to traditional project management. Agile approaches are typically used in software development to help businesses respond to unpredictability.

What is Scrum?



The Scrum Reference Card

Scrum is the most popular way of introducing Agility due to its simplicity and flexibility. Because of this popularity, many organizations claim to be “doing Scrum” but aren’t doing anything close to Scrum’s actual definition. Scrum emphasizes empirical feedback, team self-management, and striving to build properly tested product increments within short iterations. Doing Scrum as it’s actually defined usually comes into conflict with existing habits at established non-Agile organizations.

Scrum has only three roles: Product Owner, Team, and Scrum Master. These are described in detail by the Scrum Training Series. The responsibilities of the traditional project manager role are split up among these three Scrum roles. Scrum has five meetings: Backlog Grooming (aka Backlog Refinement), Sprint Planning, Daily Scrum (aka 15-minute stand up), the Sprint Review Meeting, and the Sprint Retrospective Meeting.

Many books and classes are available from a variety of competing sources of varying accuracy and quality. One place to start would be the Scrum Training Series, which uses an entertaining approach to cover the most popular way of introducing Agile to teams. You can also download the 6-page illustrated Scrum Reference Card.

What's Wrong with Traditional Approaches?

In 1970, Dr. Winston Royce presented a paper entitled “Managing the Development of Large Software Systems,” which criticized sequential development. He asserted that software should not be developed like an automobile on an assembly line, in which each piece is added in sequential phases, each phase depending on the previous. Dr. Royce recommended against the phase-based approach in which developers first gather all of a project’s requirements, then complete all of its architecture and design, then write all of the code, and so on. Royce specifically objected to the lack of communication between the specialized groups that complete each phase of work.

It’s easy to see the problems with “waterfall” methodology. It assumes that every requirement of the project can be identified before any design or coding occurs. Could you tell a team of developers everything that needed to be in a software product before any of it was up and running? Or would it be easier to describe your vision to the team if you could react to functional software? Many software developers have learned the answer to that question the hard way: At the end of a project, a team might have built the software it was asked to build, but, in the time it took to create, business realities have changed so dramatically that the product is irrelevant. Your company has spent time and money to create software that no one wants. Couldn’t it have been possible to ensure the end product would still be relevant before it was actually finished?

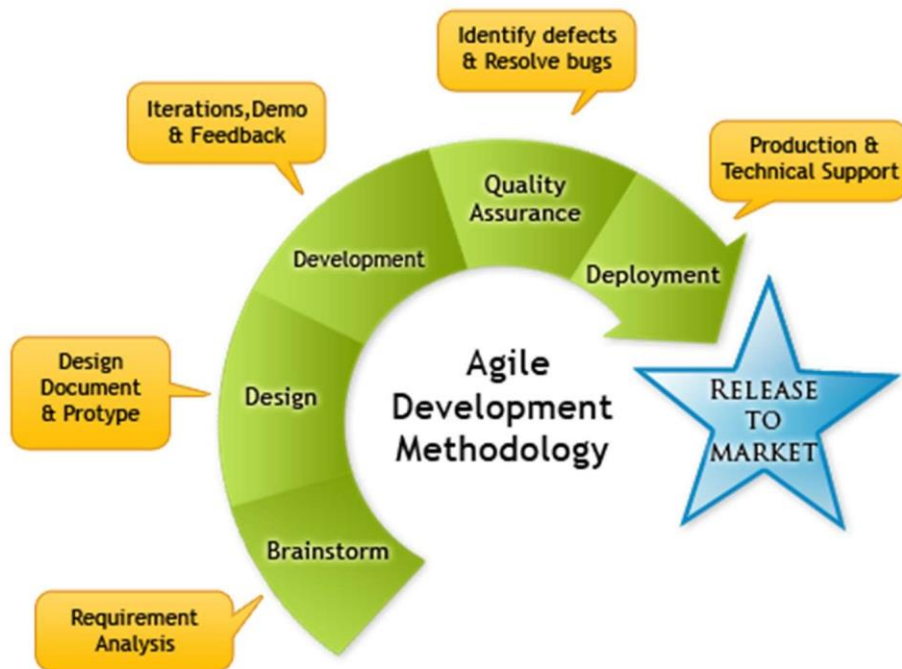
Today very few organizations openly admit to doing waterfall or traditional command and control. But those habits persist.

Why Agile?

Agile development provides opportunities to assess the direction throughout the development lifecycle. This is achieved through regular cadences of work, known as Sprints or iterations, at the end of which teams must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental.” In waterfall, development teams only have one chance to get each aspect of a project right. In an agile paradigm, every aspect of development — requirements, design, etc. — is continually revisited. When a team stops and reevaluates the direction of a project every two weeks, there’s time to steer it in another direction.

This “inspect-and-adapt” approach to development greatly reduces development costs and time to market. Because teams can develop software at the same time they’re gathering requirements, “analysis paralysis” is less likely to impede a team from making progress. And because a team’s work cycle is limited to two weeks, stakeholders have recurring opportunities to

calibrate releases for success in the real world. Agile development helps companies build the right product. Instead of committing to market a piece of software that hasn’t been written yet, agile empowers teams to continuously replan their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. Agile development preserves a product’s critical market relevance and ensures a team’s work doesn’t wind up on a shelf, never released.



Benefits of Agile Methodology

1. Delivers highest business values early on in the project
2. Promotes customer satisfaction
3. Provides customer driven approach
4. Focuses on speed of delivery
5. Provides openness and visibility to customers
6. Removes impediments in a prioritized and systematic manner
7. Improves employee retention by empowering employees and by promoting self-management, team communication, learning, and value-building

PRODUCTS

- ERP SOLUTION FOR INVETORY MANAGEMENT
- ERP APPLICATION FOR SCHOOL MANAGER
- HR MANAGERS APP
- ETC.

THANK YOU