PRACTICAL - 1

AIM: List & draw at least 7 various Software Development Life Cycle (SDLC) models and preparing the detailed case study on “Microsoft Team” which SDLC model is suitable to develop the “Microsoft Team” desktop application?

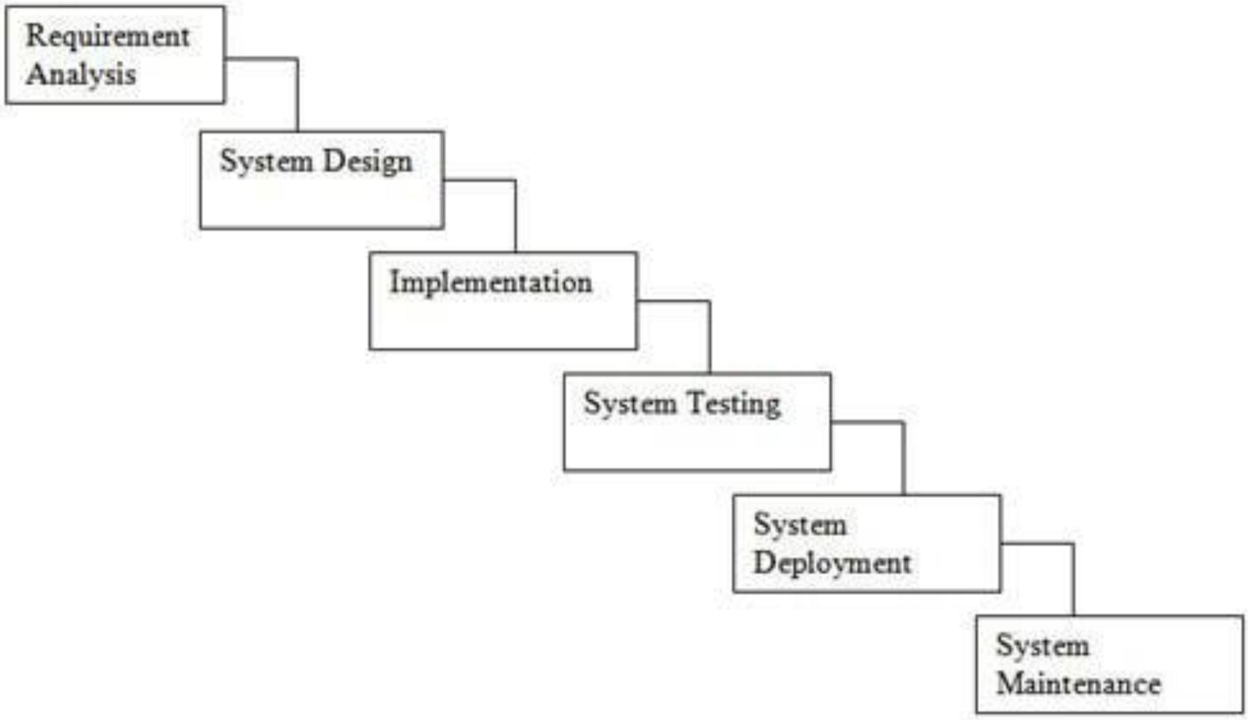
THEORY:

Software development life cycle (SDLC) models show the ways to navigate through the complex and demanding process of software building. A project’s quality, timeframes, budget, and ability to meet the stakeholders’ expectations largely depend on the chosen model.
Today, there are more than **50** recognized SDLC models in use. None of them is perfect, and each brings its favourable aspects and disadvantages for a specific software development project or a team.

Types of SDLC Models:

1. **WATERFALL MODEL:**

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 this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

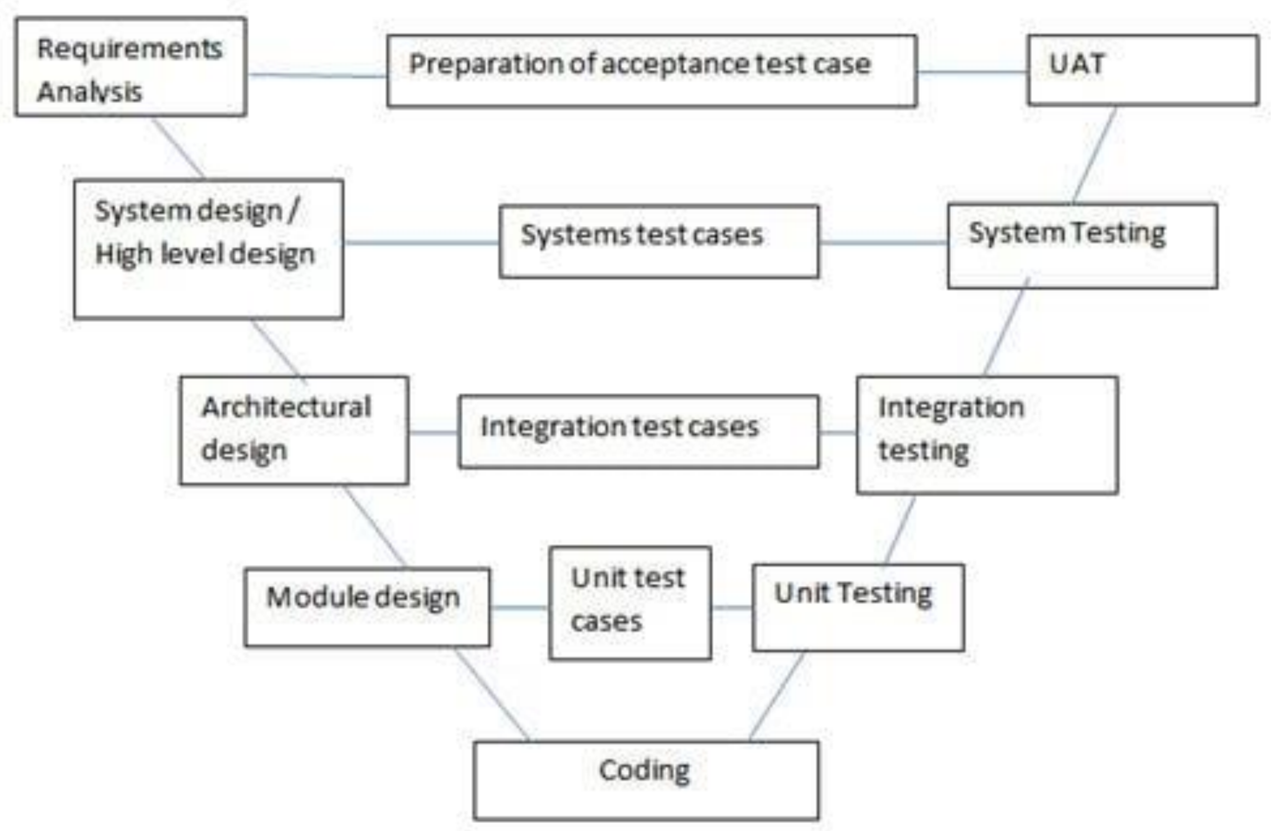


Use cases:

- 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.

2. V-MODEL:

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

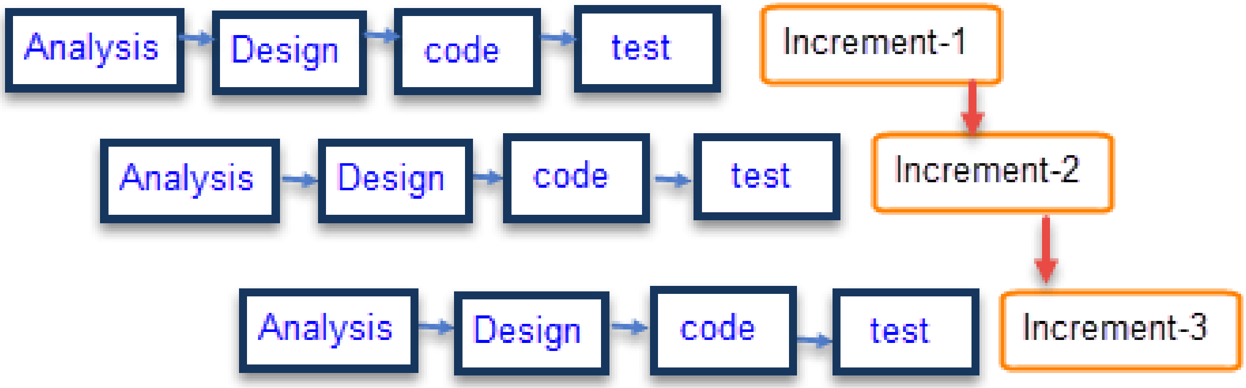


Use cases:

- Product definition is stable.
- Technology is not dynamic and is well understood by the project team.
- There are no ambiguous or undefined requirements.
- The project is short.

3. INCREMENT MODEL:

The development process based on the Incremental model is split into several iterations ("Lego-style" modular software design is required!). New software modules are added in each iteration with no or little change in earlier added modules. The development processes can go either sequentially or in parallel. Parallel development adds to the speed of delivery, while many repeated cycles of sequential development can make the project long and costly.



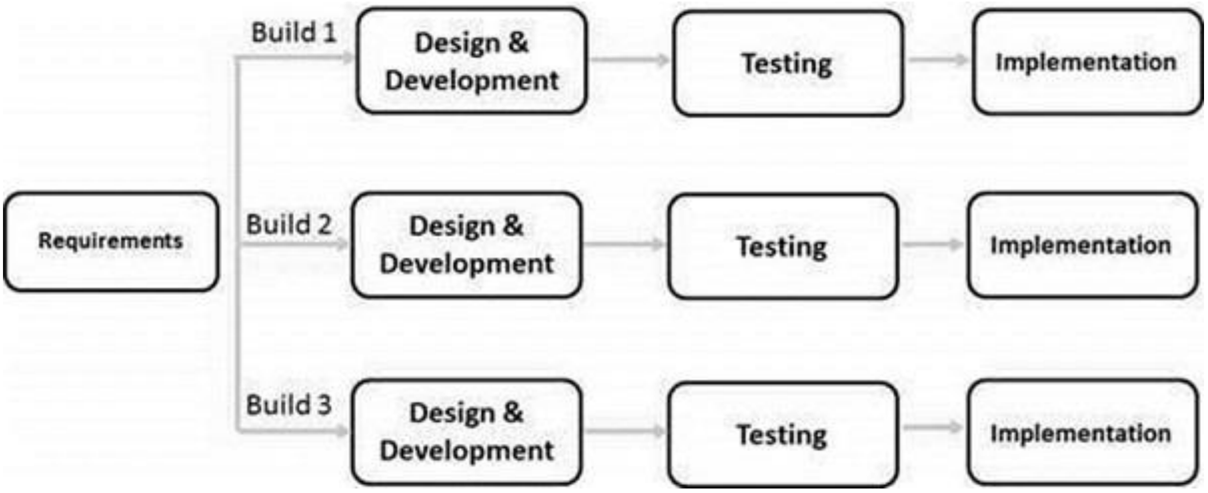
Use cases:

- Large, mission critical enterprise applications that preferably consist of loosely coupled parts, such as microservices or web services.

4. ITERATIVE MODEL:

In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.

An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

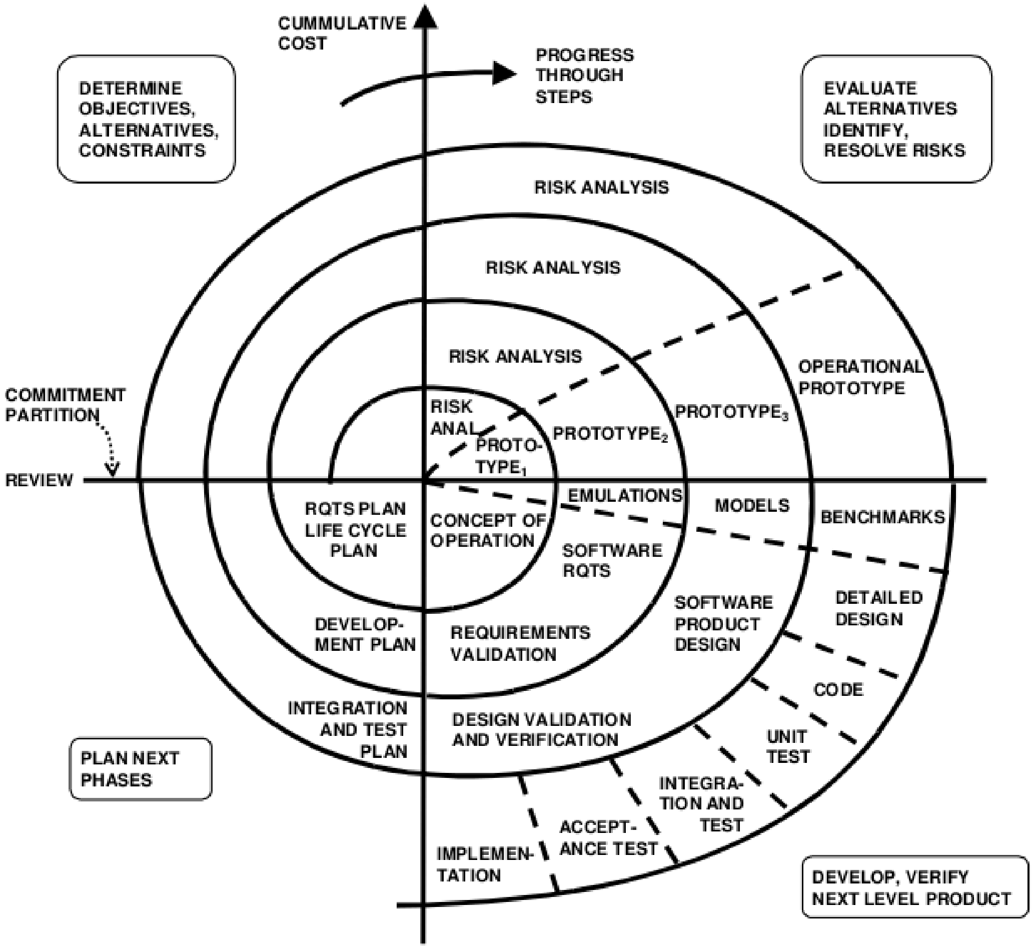


Use cases:

- Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
- There is a time to the market constraint.
- A new technology is being used and is being learnt by the development team while working on the project.
- Resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.
- There are some high-risk features and goals which may change in the future.

5. SPIRAL MODEL:

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.



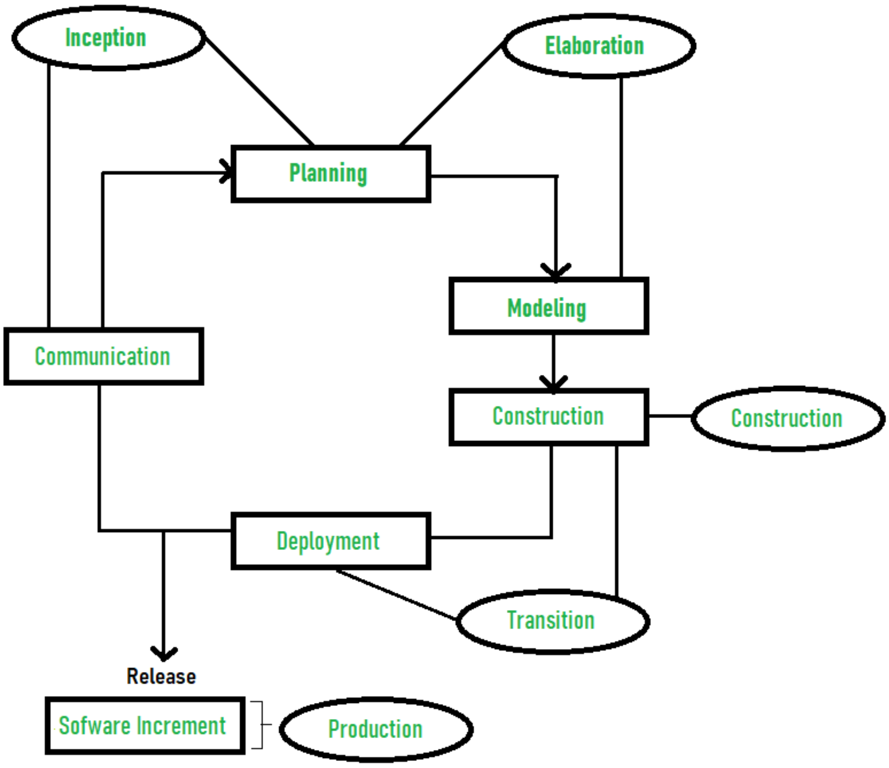
Use cases:

- When there is a budget constraint and risk evaluation is important.
- For medium to high-risk projects.
- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customer is not sure of their requirements which is usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

6. RATIONAL UNIFIED PROCESS:

The Rational Unified Process (RUP) is also a combination of linear and iterative frameworks. The model divides the software development process into 4 phases – inception, elaboration, construction, and transition. Each phase but Inception is usually done in several iterations. All basic activities (requirements, design, etc.) of the development process are done in parallel across these 4 RUP phases, though with different intensity.

RUP helps to build stable and, at the same time, flexible solutions, but still, this model is not as quick and adaptable as the pure Agile group (Scrum, Kanban, XP, etc.). The degree of customer involvement, documentation intensity, and iteration length may vary depending on the project needs.



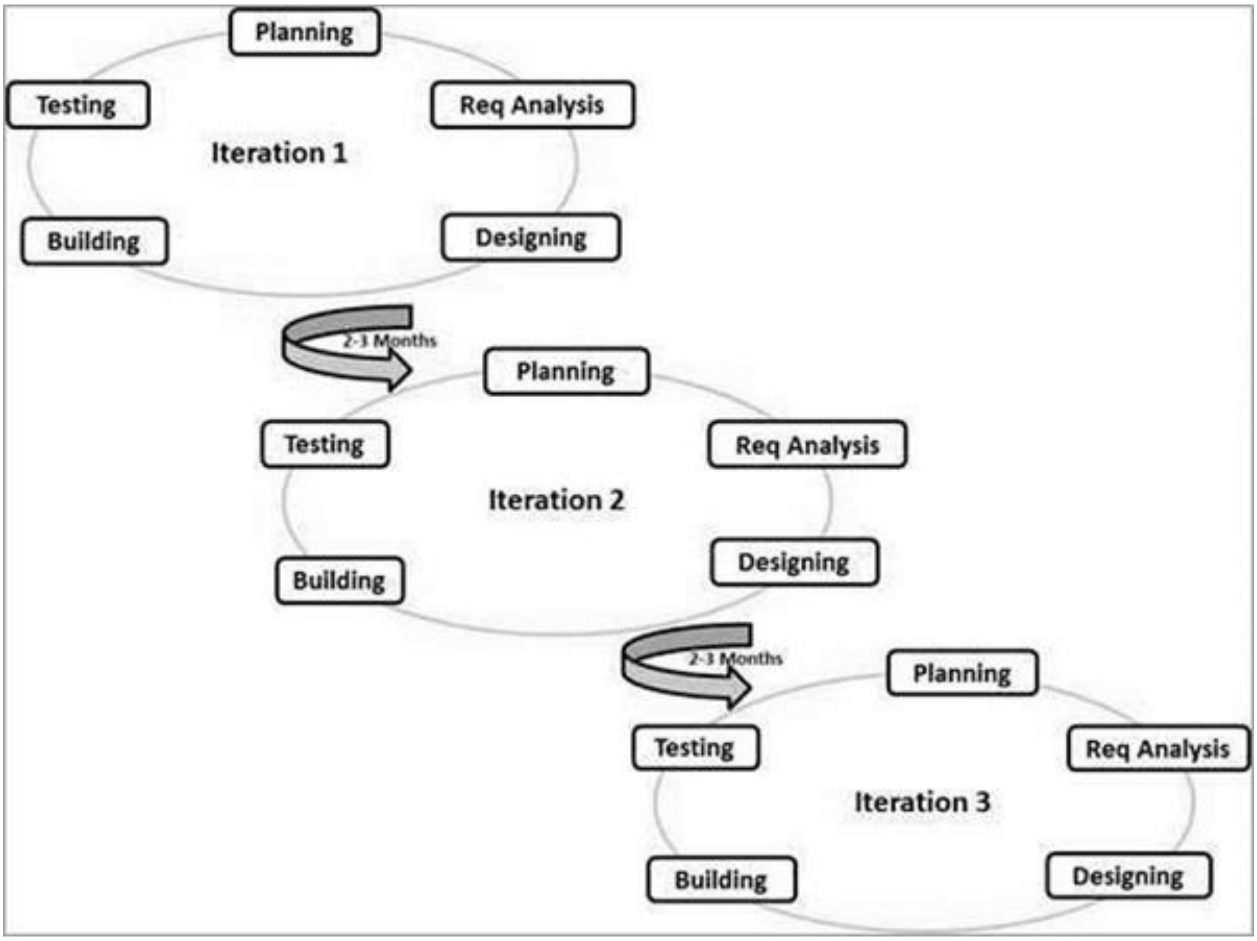
Use cases:

- Large and high-risk projects, especially, use-case based development and fast development of high-quality software.

7. AGILE MODEL:

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.



Use cases:

- Practically any start-up initiatives, when end user’s early feedback is required.
- Most mid-sized projects in custom software development were business requirements cannot be confidently translated to detailed software requirements.
- Large projects that are easy to divide into small functional parts and can be developed incrementally over each iteration.

CASE STUDY ON "MCROSOFT TEAMS"

Overview:

With the ongoing pandemic of COVID-19, mostly all the organisations were left with no option rather than adopting technologies to make sure the work is done without any barriers. Keeping in mind Microsoft Teams a collaboration and communication software emerged as vital technology to ensure not only the industrial sector but it went good with the Education purpose as well. Microsoft Teams is a tool designed to transform the way people work and interact with each other, with collaboration firmly in mind. Over the past three years, Teams has grown significantly in both new capabilities and usage, as the hub for teamwork that brings people together and fosters a culture of engagement and inclusion.

Starting September 2019, Devang Patel Institute of Advance Technology and Research migrated to Microsoft Teams to continue the Teaching-Learning process. With this migration the initial stages were a bit challenging but it went good with the time. It really helped in a few situations when asked to give advanced features of file sharing and other collaborative things.

Solution Included:

- Provided custom role-based training in step with the migration
- Teams Essential + Conferencing + Telephony
- Teams Advanced
- Provided Virtual Sessions for remote employees
- Provided continuous learning

No. of End-Users:

- 1200+

Result:

- Successful end-user adoption of teams
- Higher user satisfaction

More examples:

- Air France
- Beiersdorf

CONCLUSION

From the above theory and case study, Agile Methodology suits best for the development of "Microsoft Teams" desktop application.

PRACTICAL - 2

AIM: List at least 10 Agile Development tools for desktop and mobile application development in IT industry and prepare the detailed case study on “JIRA Tool” for Agile Development.

THEORY

1. JIRA

JIRA is a defect tracking tool which is used for Agile testing as well as project management. This tool is not only used for recording, reporting but also integrated with code development environment.

FEATURES:

- JIRA Query Language helps to create quick filters with a single click
- This agile tool helps your team become more accurate and efficient
- Reporting functionality gives team critical insight into their agile process
- Extensive reporting functionality gives your team critical insight into their agile process.
- Allows creating custom workflows of any size which is helpful to build, test, and release software

2. ZEPHYR

Zephyr is the #1 selling test management tool, providing end-to-end solutions for agile teams of all sizes. Get the flexibility, visibility, and insights you need to release better software FASTER

KEY FEATURES:

- 1-click Integration with JIRA, Confluence, Jenkins, Bamboo, and more
- Cloud, Server, and Data Centre Deployment Options
- Advanced Analytics and DevOps Dashboards
- No Annual Commitment Required

3. SPRINTS

Sprints is a tool that helps you to manage your team and product with ease. It enables you to track your progress with no hassle. This software can be used to find bottlenecks and discover ways to generate business value.

FEATURES:

- It is integrated with CI/CD tools.
- This tool helps you to get product feedback with ease.
- Allows you to work on any device and place.

- Enables the team to comment on code changes.

4. SNAGIT

Snagit is a popular screenshot capturing tool. It provides powerful tools to edit, annotate and share screenshots. It can also be used to submit and push screenshots directly.

FEATURES:

- This agile testing tool offers complete features for screen capture and video recording
- Capture videos with a simple, intuitive screen recorder
- Capture a website, record an online meeting or send feedback in an email

5. JMETER

Jmeter application is an open source agile performance testing tool. It is used to load functional test behavior and measure performance of the website.

FEATURES:

- Ability to load and performance test different applications/server and protocols
- Full featured Test IDE for fast Test Plan recording
- This agile tool offers complete portability and 100% Java purity
- Data analysis and visualization plugins offers great extensibility
- Functions can be used to offer dynamic input to test or provide data manipulation
- Easy Continuous Integration using third party libraries for tools like Maven, Gradle,and Jenkins

6. SELENIUM

Selenium is an automation agile testing tool. It aims to mimic the behavior of a real user, and as such interacts with the HTML of the application.

FEATURES:

- It is a compact Object Oriented API
- This agile tool Support for different languages like Java, Python, Ruby, Perl, PHP, and Java script
- Selenium server initializing is not required
- WebDriver finds any coordinates of any object
- It is easy tool for a WebDriver to build a keyword driven framework

7. APPIUM

Appium is an open-source and free Agile tool. It is helpful for automating mobile web, iOS, and Android and hybrid applications. Native apps are those written using Android, iOS, or Windows SDKs.

FEATURES:

- Easy process setup process
- This best agile tool supports Safari on iOS and Chrome or the built-in 'Browser' app on Android
- It can automate Native, Hybrid, and Web mobile applications
- It supports programming languages like- Java, PHP, Ruby, Python, C#, etc.
- This agile testing tool allows native, hybrid and web application testing on physical gadgets as well as on emulator or simulator.

8. BACKLOG

Backlog is an all-in-one project management tool built for developers. Agile Teams use Backlog to work with other teams for enhanced team collaboration and high-quality project delivery.

FEATURES:

- Easy bug tracking tool
- Project and issues with subtasks
- Git and SVN built-in
- Gantt Charts and Burndown charts
- Wikis
- Watchlists
- Native mobile apps
- Available both in cloud and on-premise

9. SOAP UI

Soap UI is an agile testing tool for service-oriented architectures (SOA) and REST. Its functionality includes web service inspection, invoking, development, functional testing, and load testing.

FEATURES:

1. It is open source testing tool
2. This agile tool offers Drag and Drop Test Creation
3. It allows reusing functional test cases and security scans in just a few clicks
4. It Supports Data-Driven Testing
5. Multi Environment Support
6. Allows service Simulation
7. Static Content Mocking

10. USERSNAP

User map is an Agile testing tool that allows web developers to get screenshots of bugs. This tool helps testers and developers communicate bugs easily.

FEATURES:

- Runs on every known web browsers
- This best agile tool Connects users with clients and colleagues
- Does not impede website speed
- Get visual bug reports with advanced client-side error recording
- Supports Single Page Applications

CASE STUDY ON "AGILE DEVELOPMENT WITH JIRA"

Overview:

Complaints about project management tools are an old cliché in the software development industry. They can be heard from people in many roles, in companies of all sizes, and from all countries. When listening to them, one could come to believe that, oddly, none of the more than one hundred tools existent in the market solves the problems it should.

On top of that, it's particularly interesting to note that the most popular tool in the industry, Jira Software, is notably the most infamous. Most project management tool users, both heavy and light users, have a negative opinion of Jira — weirdly, even those who have never been in direct touch with it. Several times, I have heard statements like: "I've never used Jira, but I know it sucks," and "Project management tools such as Jira are counterproductive because they slow down the team." It's almost like companies were adopting Jira just to go against the grain. Not surprisingly, though, such comments usually come from teams that follow no specific methods and processes of work and in which no one has significant experience with project management.

Agile teams use an iterative approach to break down complex projects. In the past, software development teams often failed to meet deadlines because of the sheer complexity of their work. The agile methodology for project management was developed to address the many inefficiencies present in traditional software development projects.

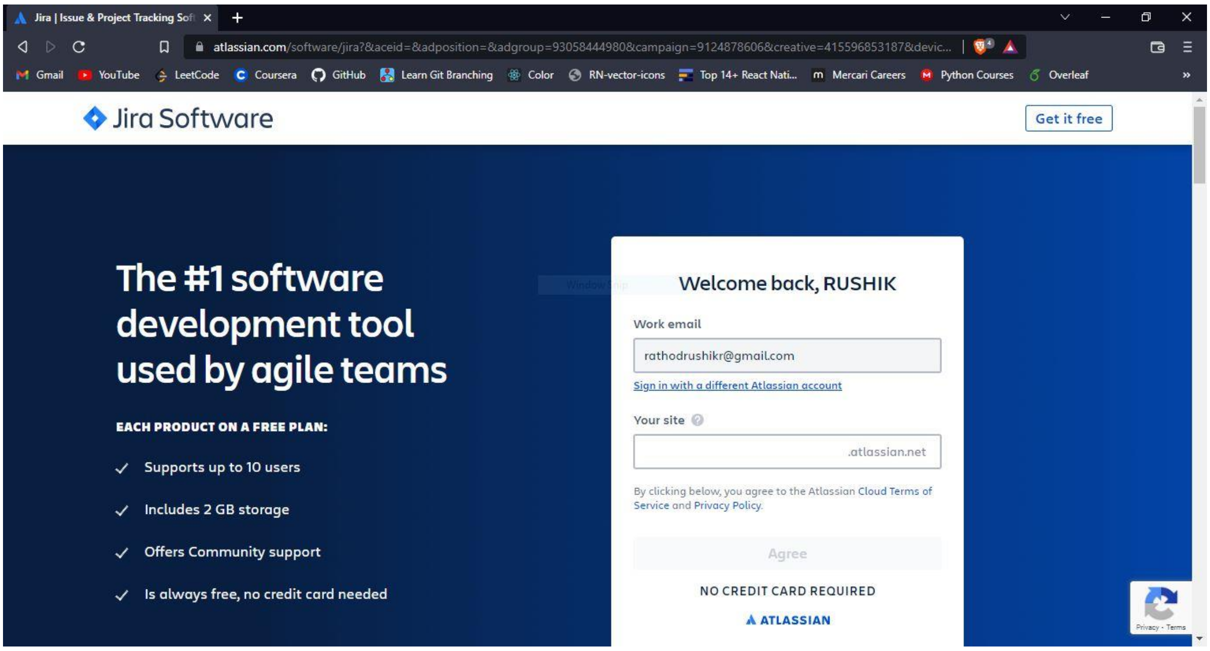
The agile methodology is based on four basic tenets:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile project managers embrace change. Since working software is the primary measure of progress for agile teams, the trajectory of the project is bound to change as feedback from clients, customers, and team members are integrated with each new release. While this might sound like it would slow the whole process down, it actually makes teams more nimble and ensures higher code quality.

JIRA…

1) Login/Sign up using email and providing website name.


Jira | Issue & Project Tracking Sof
atlassian.com/software/jira?&aceid=&adposition=&adgroup=93058444980&campaign=9124878606&creative=415596853187&devic...
Gmail YouTube LeetCode Coursera GitHub Learn Git Branching Color RN-vector-icons Top 14+ React Nati... Mercari Careers Python Courses Overleaf
Jira Software
Get it free

The #1 software development tool used by agile teams

EACH PRODUCT ON A FREE PLAN:

- ✓ Supports up to 10 users
- ✓ Includes 2 GB storage
- ✓ Offers Community support
- ✓ Is always free, no credit card needed

Welcome back, RUSHIK

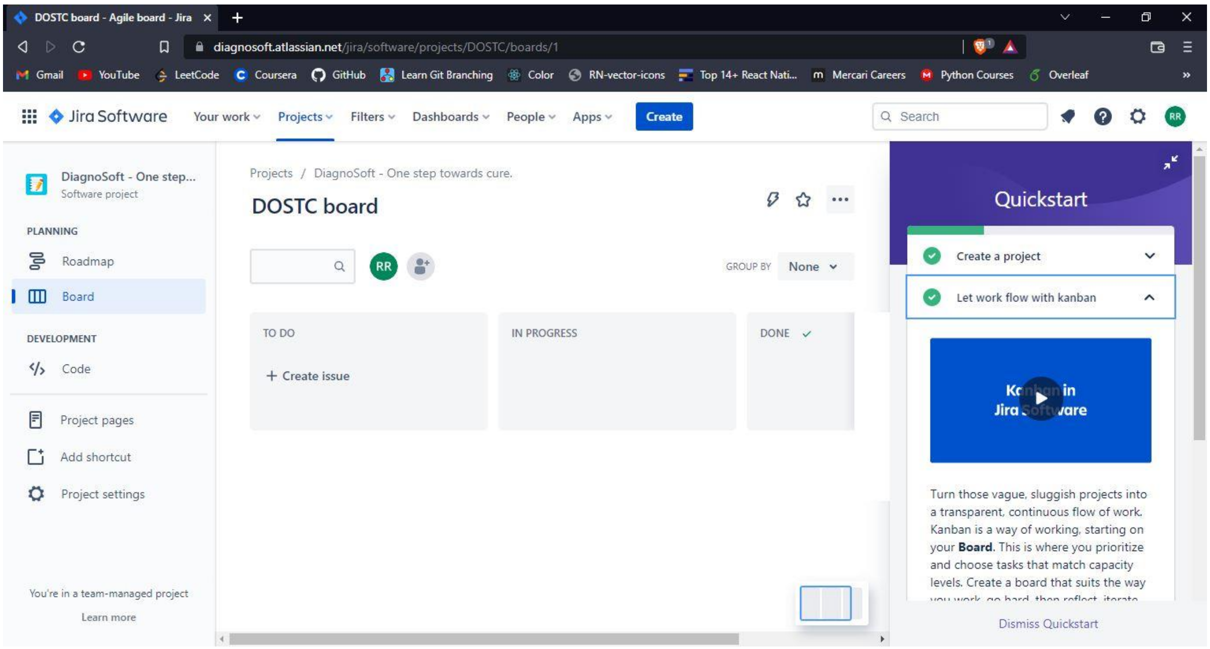
Work email: rathodrushikr@gmail.com

Sign in with a different Atlassian account

Your site: .atlassian.net

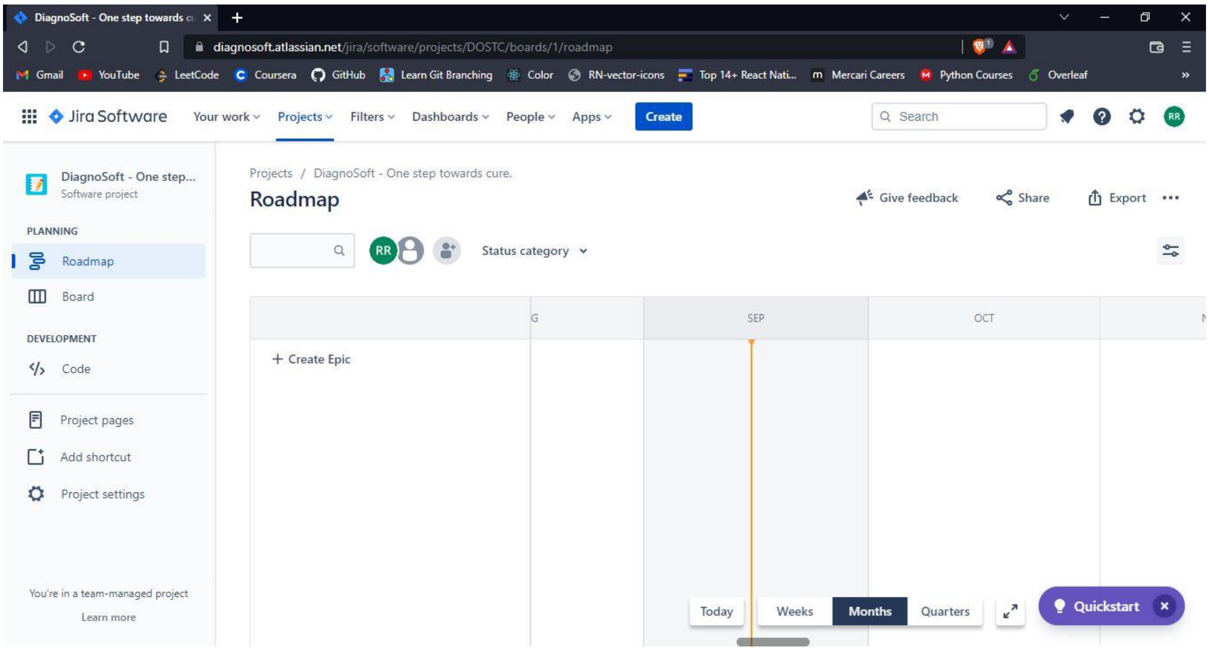
By clicking below, you agree to the Atlassian Cloud Terms of Service and Privacy Policy.

2) DiagnoSoft board.

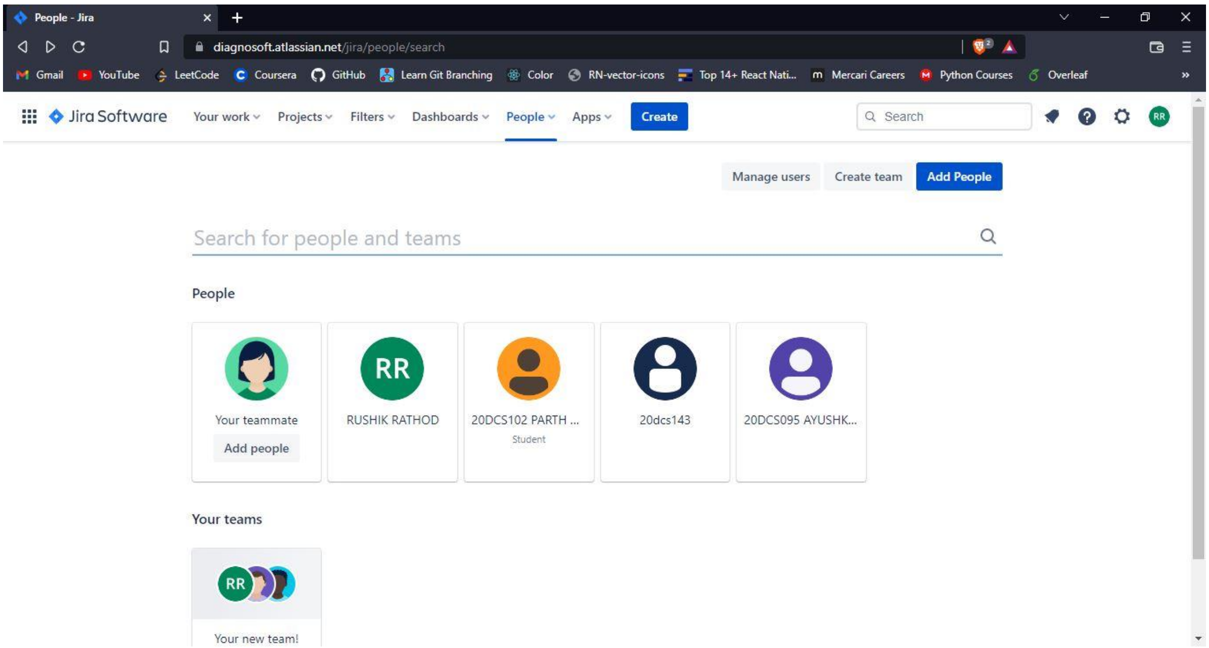

DOSTC board - Agile board - Jira
diagnosoft.atlassian.net/jira/software/projects/DOSTC/boards/1
Gmail YouTube LeetCode Coursera GitHub Learn Git Branching Color RN-vector-icons Top 14+ React Nati... Mercari Careers Python Courses Overleaf
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
RR
DiagnoSoft - One step...
Software project
PLANNING
Roadmap
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / DiagnoSoft - One step towards cure.
DOSTC board
RR
GROUP BY None
TO DO
Create issue
IN PROGRESS
DONE ✓
Quickstart
Create a project
Let work flow with kanban
Kanban in Jira Software
Dismiss Quickstart

Turn those vague, sluggish projects into a transparent, continuous flow of work. Kanban is a way of working, starting on your **Board**. This is where you prioritize and choose tasks that match capacity levels. Create a board that suits the way you work, go hard, then reflect, iterate

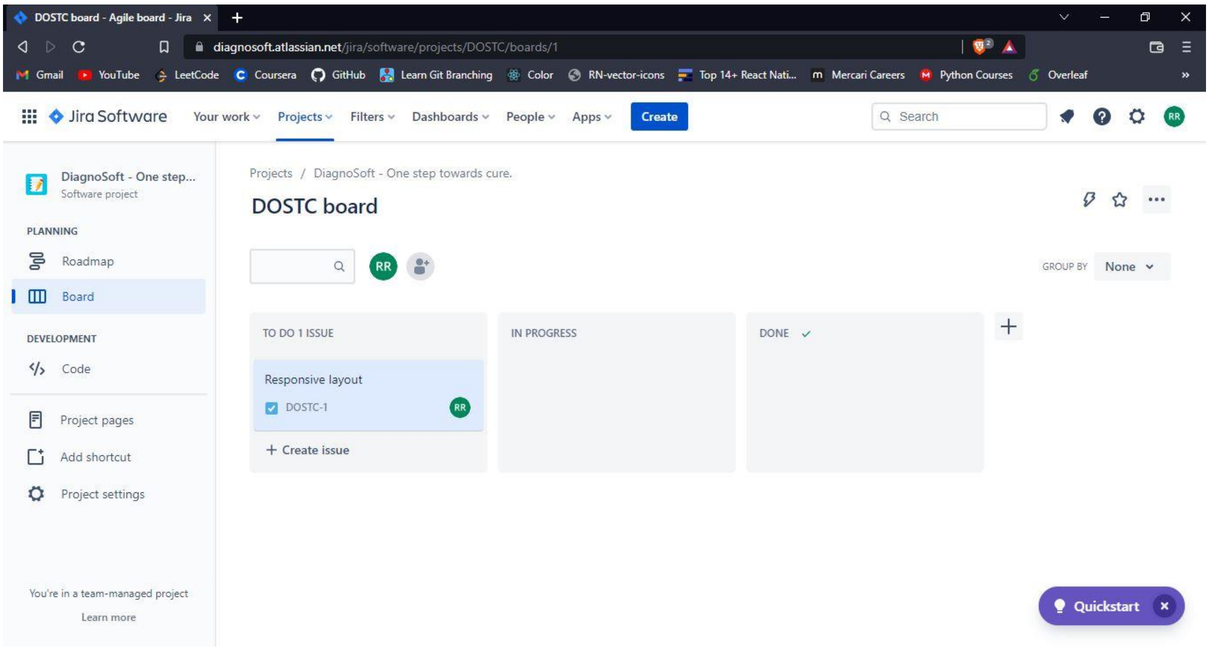
3) Roadmap page.



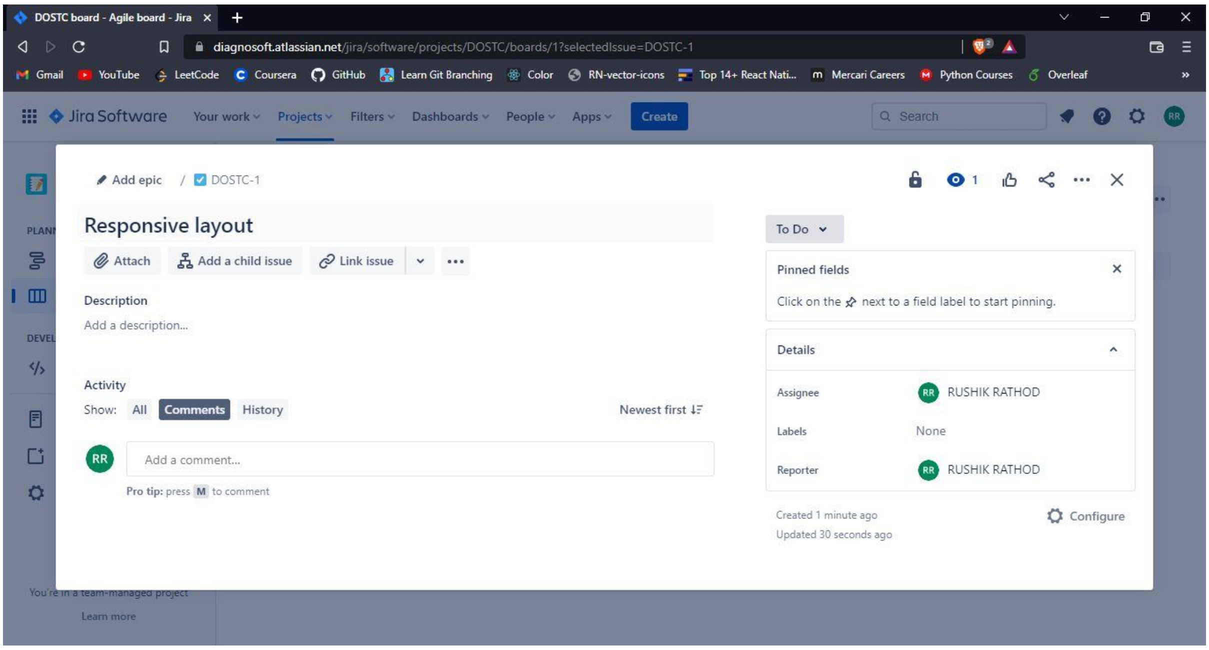
4) Creating a team.



5) Issue created.



6) Assigning issue to a team member.


DOSTC board - Agile board - Jira
diagnosoft.atlassian.net/jira/software/projects/DOSTC/boards/1?selectedIssue=DOSTC-1
Jira Software
Add epic / DOSTC-1
Attach
Add a child issue
Link issue
To Do
Pinned fields
Click on the next to a field label to start pinning.
Details
Configure
Activity
Show: All Comments History
Newest first
Add a comment...
Pro tip: press M to comment

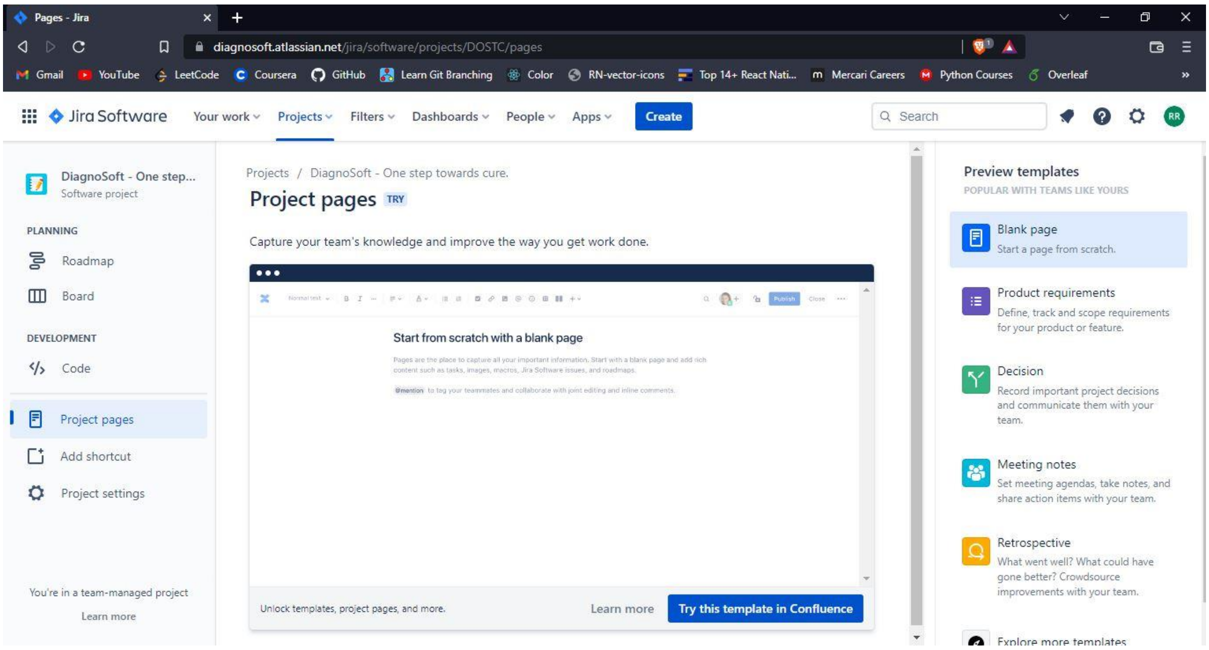
Responsive layout

Description
Add a description...

Assignee: RUSHIK RATHOD
Labels: None
Reporter: RUSHIK RATHOD

Created 1 minute ago
Updated 30 seconds ago

7) Project pages.



Capture your team's knowledge and improve the way you get work done.

Start from scratch with a blank page

Pages are the place to capture all your important information. Start with a blank page and add rich content such as tasks, images, macros, Jira Software issues, and roadmaps.

@mention to tag your teammates and collaborate with joint editing and inline comments.

Unlock templates, project pages, and more.

Blank page
Start a page from scratch.

Product requirements
Define, track and scope requirements for your product or feature.

Decision
Record important project decisions and communicate them with your team.

Meeting notes
Set meeting agendas, take notes, and share action items with your team.

Retrospective
What went well? What could have gone better? Crowdsource improvements with your team.

Jira has adopted agile principles into every facet of their software. Agile teams can quickly and easily navigate charts, long-term goals (epics), and tasks. Tracking issues and spotting bugs can also be done automatically by Jira Software.

CONCLUSION

Jira is a powerful project management system with planning, tracking, releasing, and reporting all in one place. Teams are able to access information about tasks, productivity, bugs, and code quality in one location.

PRACTICAL - 3

AIM: List at least 5 software development planning tools and prepare the detailed case study
of Risk Analysis & Management (i.e. Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation.) on "DiagnoSoft" mobile application

THEORY

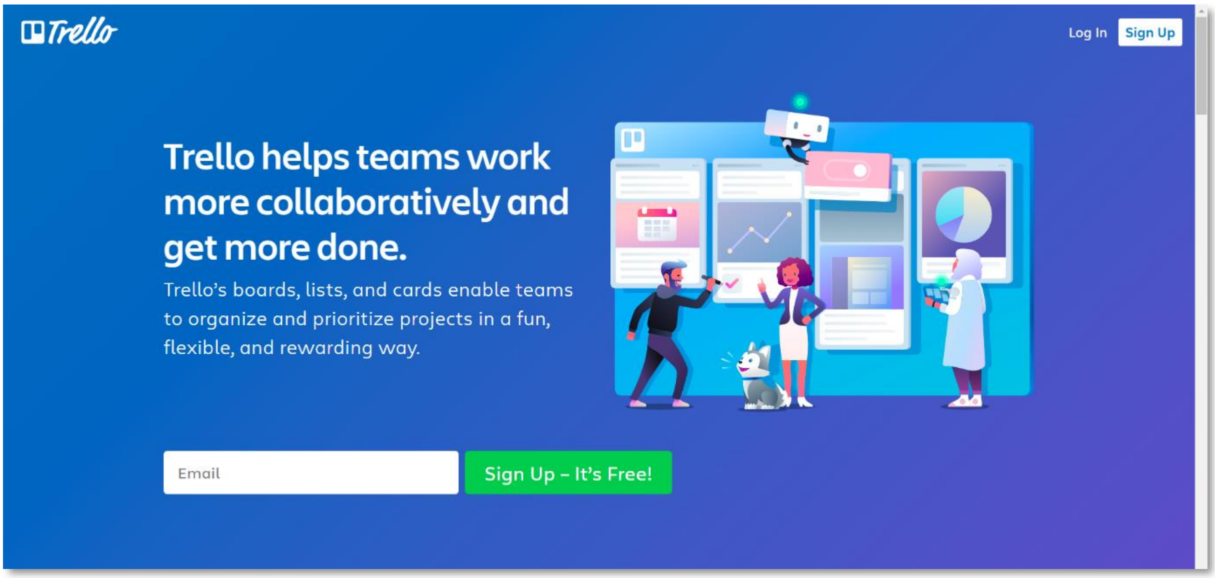
Need of Project Planning:

Software development is a sort of all new streams in world business, and there's next to no involvement in structure programming items.

Most programming items are customized to accommodate customer's necessities. The most significant is that the underlying technology changes and advances so generally and rapidly that experience of one element may not be connected to the other one.

All such business and ecological imperatives bring risk in software development; hence, it is fundamental to manage software projects efficiently

1. TRELLO

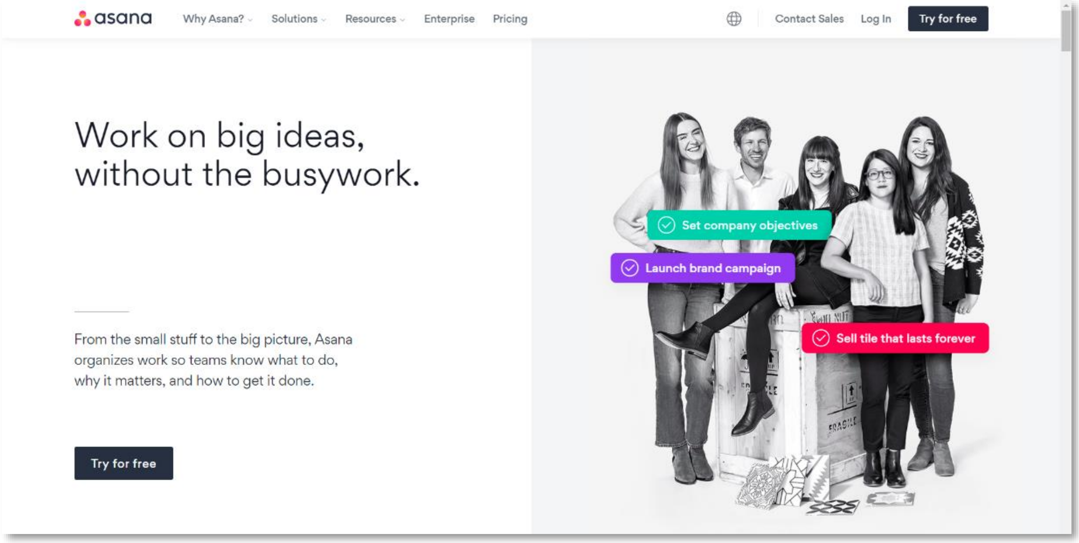


Trello helps teams work more collaboratively and get more done.

Trello's boards, lists, and cards enable teams to organize and prioritize projects in a fun, flexible, and rewarding way.

Trello allows you to create boards and fill them with different lists. I went for a simple Kanban-based board and created lists that correspond with statuses (in progress, to do etc.) Trello is a tool that would allow me to create more granular tasks but I've decided to use checklists that are built inside the cards to track smaller tasks. You can also add attachments to cards, making it easier to communicate new iterations or getting feedback. Adding due dates to particular cards is also handy when you want to plan a project.

2. ASANA

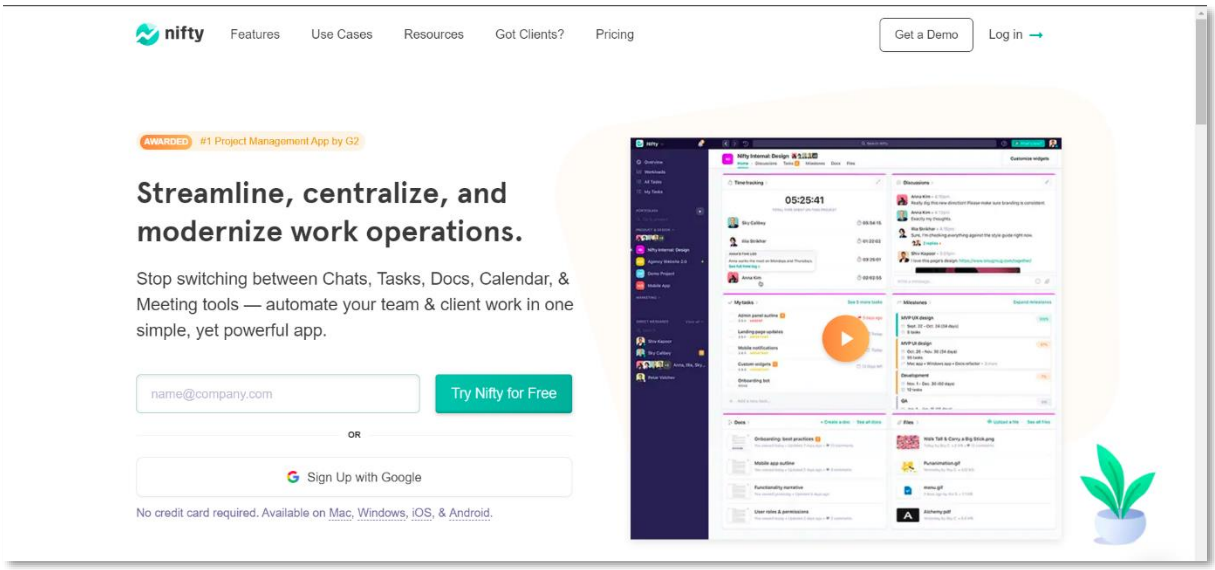


Work on big ideas, without the busywork.

From the small stuff to the big picture, Asana organizes work so teams know what to do, why it matters, and how to get it done.

Asana is a popular tool for managing projects of various sizes, and while the free version comes with limited features, it seems to be perfectly suitable for smaller projects. The plan of your project can be displayed as a list, a board or a calendar (see below). The latter view is especially useful when planning milestones and deadlines. You can see how much time there actually is between different due dates. Keep in mind, however, that setting start dates for tasks is not available in the free plan.

3. NIFTY



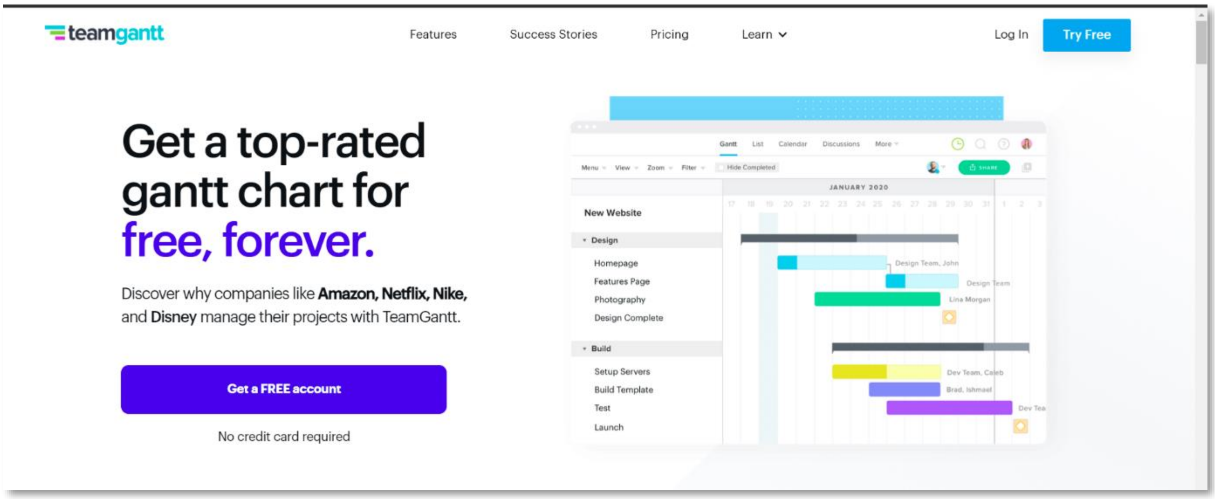
Streamline, centralize, and modernize work operations.

Stop switching between Chats, Tasks, Docs, Calendar, & Meeting tools — automate your team & client work in one simple, yet powerful app.

No credit card required. Available on Mac, Windows, iOS, & Android.

The creators of Nifty pride themselves on building an app that covers project management and team collaboration. The collaboration aspect is supported by the fact that a team chat is a part of this app. What about the project management part? You can plan your project starting with a list of tasks, or map your milestones on a Gantt chart.

4. TEAMGANTT

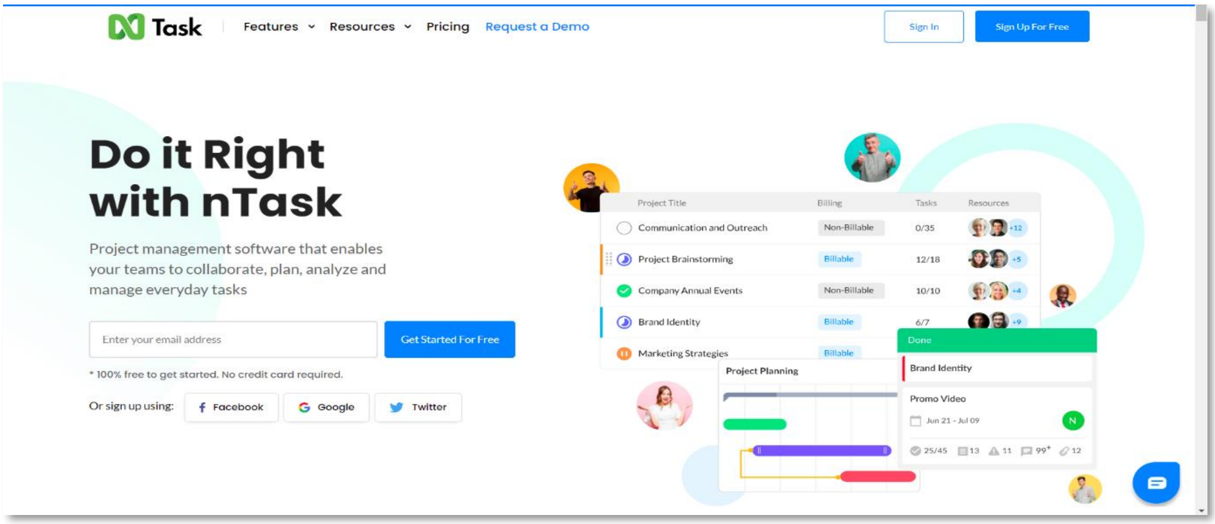


Get a top-rated gantt chart for free, forever.

Discover why companies like **Amazon, Netflix, Nike,** and **Disney** manage their projects with TeamGantt.

Out of the box you'll notice that TeamGantt will not leave you hanging. The onboarding experience will help you tremendously, especially if you're new to project planning or gantt charts in general. As the name of this software would suggest, creating Gantt charts is the core functionality here. The process of building and editing charts is very straightforward, you can also assign people to particular tasks

5. NTASK



Do it Right with nTask

Project management software that enables your teams to collaborate, plan, analyze and manage everyday tasks

* 100% free to get started. No credit card required.

If you're looking for a tool for planning and managing a series of projects, nTask could just be right for you. Of course, it will also be suitable for single project purposes. It's just that nTask is really good at providing you with the big picture of all your ongoing projects. When it comes to planning a single project, you can create a list of tasks or a simple Gantt chart. The assignments can also be viewed as a grid.

CASE STUDY ON “RISK ANALYSIS & MANAGEMENT IN “DiagnoSoft”

Overview:

DiagnoSoft is a Machine Learning based Android application made using cross-platform React Native framework. It allows users to predict the whether or not a person is having COVID-19 and brain-tumor by just uploading the X-ray image. DiagnoSoft is being developed with a view to provide the instant result with just one click taking the great advantage of emerging technology. Since development of such great app includes high risk factors which we will analyse here.

Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.

A risk is a probability that some adverse circumstance will occur.

- Project risks affect schedule or resources
- Product risks affect the quality or performance of the software being developed
- Business risks affect the organisation developing or procuring the software

A risk management process contains 4 major steps.

- Risk identification - Identify project, product and business risks
- Risk analysis - Assess the likelihood and consequences of these risks
- Risk planning - Draw up plans to avoid or minimise the effects of the risk
- Risk monitoring - Monitor the risks throughout the project.

RISK ANALISIS AND MANAGEMENT ON DiagnoSoft

1. Risk Identification

From my perspective, in Risk Identification, there is some risk that can affect DiagnoSoft and that are:

- Business Impact: There are some similar kind of applications available in the market which are providing more accuracy and best user experience.
- Technology to the limit: the Technology is latest or not.
- Development environment: If the development environment changes then the application may get affected.

2. Risk Projection

RISK	CATEGORY	PROBABILITY	IMPACT
Other Applications that are similar to DiagnoSoft	BU	60%	4
Lack of Technology tools	DE	40%	2
Application not meet expectations	TE	20%	1

3. Risk Refinement

We can use the risk Identification list that are made on the basis of the risk item checklist that is built on risk analysis questions. So further that question will be get sub question like for example we have risk in Lack of technology tools so there is some sub question like what tools and what is the cost? and risk impact is on level 2 so risk has to be solved.

4. Risk Mitigation

From the Risks that we have discussed above in in risk projection:

- Other Applications that are similar to DiagnoSoft so we can avoid this Risk for Risk mitigation.
- Lack of Technology tools so we can accept the Risk and find the best employee that can work on any environment.
- Application not meet expectations so for this Risk we can reduce the Risk.

CONCLUSION

In this practical, we learned about different software development planning tools. Then we explored risk analysis and risk management. We performed case study for DiagnoSoft the on same topic.

PRACTICAL - 4

AIM: List at least 10 software design principles & online/offline tools for software development process and Draw the UML diagram for Microsoft Windows 10 operating system.

THEORY Here we will be discussing about a number of software design principles and also some of the tools for software development process.

Software Design Principles:- The first five principles form the basics of the software development principles and are also known as "S.O.L.I.D principles".

1. **Single Responsibility Principle (SRP):-** It is a software engineering principle that states that a class should have only one reason to change. In other words, it must have only one responsibility.Here, we are talking about cohesion. All elements in given class structures or modules should have a functional affinity to one another. By clearly defining your class's responsibility, you increase its cohesiveness. This is the first principle denoted by **"S".**

2. **Open/Closed Principle (OCP):-** The principle says that you should be able to change the behavior of a class without modifying it. Therefore, you can extend the class's behavior through composition, interface, and inheritance. However, you cannot open it for minor modifications. This is the second principle denoted by **"O".**

3. **Liskov Substitution Principle (LSP):-**The LSP principle mainly focusses on the degree of use of the Inheritance in a software. While inheritance is beneficial, it is advisable to use it contextually and moderately. The principle strives to prevent cases where classes are extended only through common things. You need to consider the pre-conditions and post-conditions of a class before performing inheritance. This is the third principle denoted by **"L".**

4. **Interface Segregation Principle (ISP):-** ISP prefers many specific interfaces to a general interface. The goal is to have finely grained and client-specific interfaces. You need to enhance cohesion in interfaces and develop modules with few behaviors. Interfaces that have many behaviors are hard to maintain and evolve. So, they should be completely avoided. This is the fourth principle denoted by **"I".**

5. **Dependency Inversion Principle (DIP):-** The fifth and the final principle of SOLID is Dependency Inversion Principle and is denoted by **"D".** The principle asserts that programmers should depend on abstractions and not on concrete classes. This can be broken down to 2 parts as: We can break it into two:

- ✓ High-level modules need to be independent of low-level ones. Both should depend on abstractions
- ✓ Abstractions should be independent of details. Details should depend on abstractions.

Now we look at some of the important principles other than SOLID .

6. **Keep It Simple :-** Ensuring that the program coding is simple and very easy to understand is very much important. The code shouldn't give a new person hard time to figure out how things are working. Methods and functions should be as brief as possible but also easily understandable. Each of them should solve only one problem or two at max. Also the project code should not have a lot of conditions (simple and nested conditions). Optimizing the conditions would help understanding and finding bugs easily for the QA team.

7. **You Aren't Gonna Need It (YAGNI):-** Most programmers fall into the pit of trying to implement all the functionalities at once, from the word go. In the end, some or most of these functionalities become useless. Start by adding just a few methods to a class. After that, as your project starts taking shape and new demands arise, you can add more functionalities. That way, you'll achieve a lean development software. YAGNI saves time, efforts, and costs that you would have wasted in trying to understand or debug the code.

8. **Measure Twice and Cut Once:-** The development life cycle's requirement stage usually introduces more than 50% coding issues if not done well. Therefore a systematic approach should be developed. It is very much important to double check all the project requirements so that one does not miss any important section (feature) OR does not accidentally add too much. After that, make blueprints that will guide the whole process to achieve high-quality coding throughout. Always test your project from basics to ensure everything is fine. This principle gives much more predictable outcomes, especially if the project's cost is already high. You'll save yourself headaches that come with deleting or adding code lines to meet requirements.

9. **Don't Repeat Yourself (DRY):-** When writing your code, don't repeat yourself. That is, avoid copy-pasting your code in different places. Otherwise, future maintenance will be difficult. The reason is that you will have to make changes to the coding in those various places. Those changes will further necessitate changes in the tests to make the results click green. All of that will need more time, effort, and money. To avoid such a pitfall, you can extract a common logic into functions. Additionally, if there are any manual works that you can automate, do so to keep your code lean. For software development, the above steps will help in the code re-usability without having to repeat it.

10. **Least Astonishment:-** The principle of least astonishment says that it is advisable to design a feature that doesn't have a high-astonishment factor. Your system's components should behave in a way that end-users expect. Therefore, your project's outcomes will be profitable only if they are obvious, predictable, and consistent. Otherwise, users will shy from using features or structures that astonish, surprise, or confuse them. You are making software products for people to use. Thus, you'll reap a lot by designing user-friendly features. Strive to match human beings' mental models, experience, and expectations. Remember, you have to capture the user's attention as quickly as possible. As we know, the current users' attention span has plummeted.

SOFTWARE DEVELOPMENT PROCESS TOOLS:

1. **Embold:-** Fixing bugs before deployment saves a lot of time and energy in the long run. Embold is a software analytics platform that analyses source code and uncovers issues that impact stability, robustness, security, and maintainability.

   Features:

- ✓ With the Embold plugins, you can pick up code smells and vulnerabilities as you code, before making commits.
- ✓ Unique anti-pattern detection prevents the compounding of unmaintainable code.
- ✓ Integrate seamlessly with Github, Bitbucket, Azure, and Git and plugins available for Eclipse and IntelliJ IDEA.
- ✓ Get deeper and faster checks than standard code editors, for over 10 languages.

2. **Linx:** Linx is a low code IDE and server. IT pros use Linx to quickly create custom automated business processes, integrate applications, expose web services and to efficiently handle high workloads.

   Features:-

- ✓ Easy-to-use, drag-and-drop interface
- ✓ Over 100 pre-built functions and services for rapid development
- ✓ One-click deployment to any local or remote Linx Server directly from the IDE
- ✓ Input and outputs include nearly any SQL & NoSQL databases, numerous file formats (text and binary) or REST and SOAP Web services
- ✓ Live debugging with step through logic
- ✓ Automate backend processes via timer, directory events or message queue or expose web services, and call APIs via HTTP requests

3. **Studio 3T:-** Studio 3T for MongoDB helps you to build queries fast, generate instant code, import/export in multiple formats, and much more.

   Features:-

- ✓ Query MongoDB faster with our Visual Query Builder, IntelliShell, or SQL Query tool.
- ✓ Our Data Masking tool enables data compliance and bolsters security with powerful field-level data obfuscation.
- ✓ Import to MongoDB from JSON, CSV, BSON/mongodump, and SQL, and get a preview of your output documents as you make changes.
- ✓ Migration from MongoDB to SQL (or vice versa) has never been easier with our Migration tools."

4. **Kite:-** Kite is IDE for Software Development that automatically completes multiple line codes. This editor supports more than 16 languages. It helps you to code faster with no hassle.

   Features:-

- ✓ It offers Software Development documentation.
- ✓ This editor provides a function signature as you type.
- ✓ You will get a tooltip on mouse hover.
- ✓ Provides support in email.
- ✓ Uses machine learning models for Software Development language.
- ✓ Also it is a free to use open source tool.

5. **NetBeans:-** NetBeans is an open source and a free software development tool written in Java that develops world-class web, mobile, and desktop applications easily and quickly. It uses C / C++, PHP, JavaScript, Java etc.

   Features:-

- ✓ Support for fast & smart code editing.
- ✓ Easy & Efficient Project Management process.
- ✓ Rapid User Interface Development.
- ✓ Helps to write bug-free code.
- ✓ NetBeans IDE offers superior support for C/C++ and PHP developers.

6. **Cloud9 IDE:-** Cloud9 IDE is an online integrated software development environment. It supports many programming languages like C, C++, PHP, Ruby, Perl, Python, JavaScript and Node.js.

   Features:-

- ✓ Allows to clone entire development environment.
- ✓ Built-In Terminal for command-line wizard.
- ✓ Code Completion suggestions helps software developers to code faster and avoid typos.
- ✓ The Debugger helps developers to set breakpoints, and inspect variables of any JS/Node.js app.
- ✓ Simply drag any file or Terminal to create multiple split views.
- ✓ Developers can select an extensive set of default Runners to execute app, such as Ruby, Python, PHP/Apache.

7. **Atom:-**Atom is a solid all-around text-editor. It is fully free and open source. It can be customized to do anything but without a need of modifying the config file.

   Features:-

- ✓ Atom works across many popular operating systems like OS X, Windows, or Linux.
- ✓ It helps developers to write code faster with a smart, flexible autocomplete.
- ✓ Easily browse and open whole project or multiple projects in one window.
- ✓ It is possible to split Atom interface into multiple panes to compare and edit code across files.

- ✓ Find, preview, and replace text type in a file or across the entire project.

8. **GitHub:-** GitHub is a powerful collaboration tool and development platform for code review and code management. With this GitHub, the users can build applications and software, manage the projects, host the code, review the code etc.

Features:-

- ✓ With GitHub, developers can easily document their code and can host the same from the repositories.
- ✓ GitHub's project management tools help its users to stay aligned, co-ordinate easily and get their task done accordingly.
- ✓ Few features of GitHub that make it a useful tool are its code security, access control among the team members, integration with other tools etc.
- ✓ Few developers use GitHub for experimenting new programming languages in their personal projects.
- ✓ GitHub can be hosted on servers and on a cloud platform. It runs on Windows and Mac OS.
- ✓ GitHub is free for open source projects and public use. For developers it is charged based on different criteria and services requested.

9. **Bitbucket:-** Bitbucket is a distributed, web-based version control system that is used for collaboration between software development teams (code and code review). It is used as a repository for source code and development projects.

Features:-

- ✓ Useful features of Bitbucket that makes it a powerful tool are its flexible deployment models, unlimited private repositories, code collaboration on steroids etc.
- ✓ Bitbucket supports few services like code search, issue tracking, Git large file storage, bitbucket pipelines, integrations, smart mirroring etc.
- ✓ Using Bitbucket, one can organize the repositories into the projects with which they can focus easily on their goal, process or product.
- ✓ To rationalize the development process of any software it can integrate into the prevailing workflow.
- ✓ Bitbucket offers a free plan for 5 users with unlimited private repositories.

10. **JIRA:-**Jira is the most popular software development tool that is used by agile teams for planning, tracking and releasing the software.

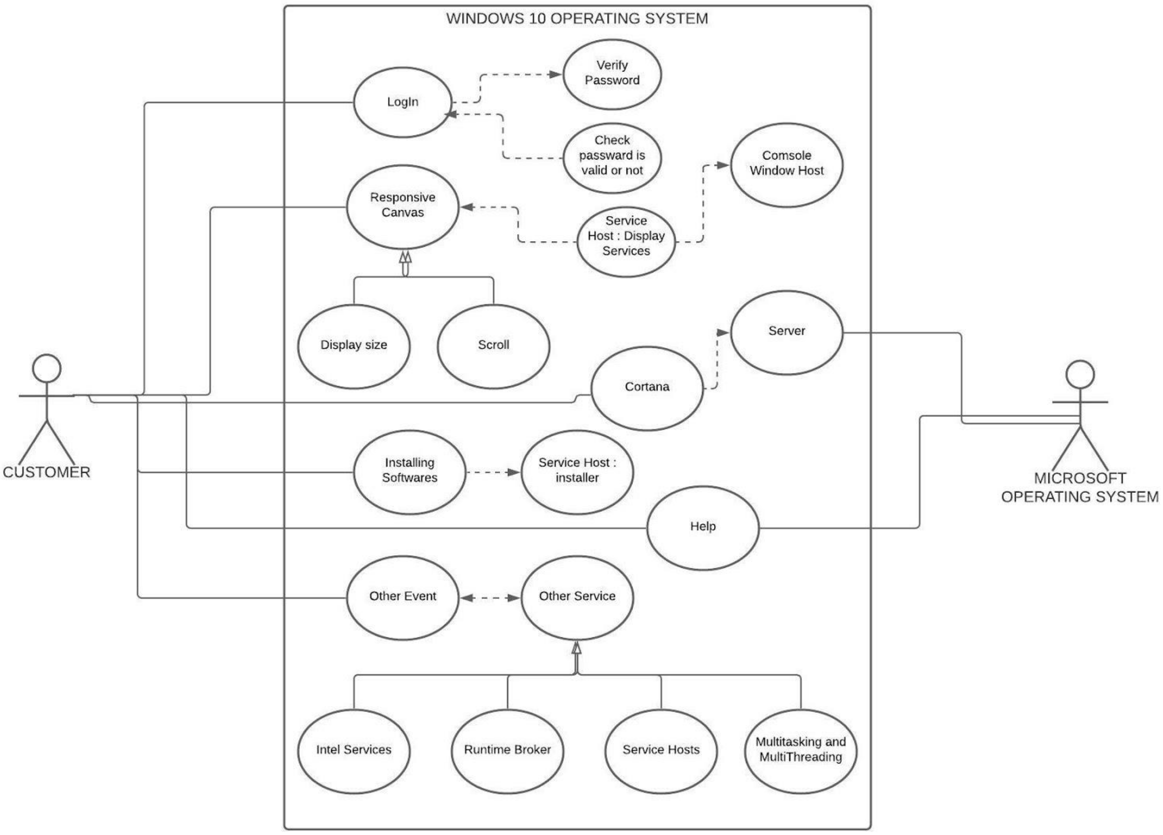
Features:-

- ✓ This tool is customizable and also has some prevailing features that are used in every development phase.
- ✓ Using Jira, we can accomplish the work in progress, generate reports, backlogs etc.

- ✓ Few other important features of Jira software are Scrum boards, Kanban boards, GitHub integration, Disaster recovery, Code Integration, Portfolio Management, Sprint Planning, Project Management etc.
- ✓ Jira works for Windows and Linux/Solaris operating systems.

UML DIAGRAM FOR WINDOWS 10 OPERATING SYSTEM

UML, short for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process.



CONCLUSION:

In this practical, we learned about different software development principles to keep in mind during a software development planning. Also we have seen some of the most popularly used software development tools which are available online as well as application software. Also at the end we have learnt about UML diagrams, how to prepare them and also seen the UML diagram of windows 10 operating system.

PRACTICAL - 5

AIM: Design the Software Requirement Specification (SRS) document on Microsoft Windows 10 Operating System in IEEE format only. Draw the UML diagram. (Diagram must include 1. Use Case, 2. Activity, 3. State, 4. Sequence, 5. Data Flow (DFD) 6. Class for Microsoft Windows 10 Operating System.)

Software Requirements Specification

For

Windows 10

Version 1.0 approved

Prepared by Rushik Rathod
20DCS103

Devang Patel Institute of Advance Technology and Research

02/09/2022

Table of Contents

Introduction

Purpose

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015. It is built on the Windows NT kernel and follows Windows 8. One of the primary aims of Windows 10 is to unify the Windows experience across multiple devices, such desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows 10 Mobile alongside Windows 10 to replaces Windows Phone – Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

Document Conventions

When you read this manual, certain words are represented in different fonts, typefaces, sizes, and weights. This highlighting is systematic; different words are represented in the same style to indicate their inclusion in a specific category. The types of words that are represented this way include the following:

o Figure caption is written in size 9.

o Regular text conventions are followed as per the IEEE Recommended Practice for

Software Requirements Specifications standard 830-1998.

o This style indicates that the program is an end-user application (as opposed to system software). For example: Use ***Chrome*** to browse the Web.

o This style indicates that the Products that uses windows For example: *HoloLens,phones*

o This style indicates that Keys: Power, Volume up

o For Reference Links : *https://windows10.org*

Intended Audience and Reading Suggestions

This document is intended can be referred by Developer, Project Manager, Marketing Staff, users, testers and documentation writers.

For a generalized reader, the SRS need to be read top-down.

The Project Manager can refer to point 2 (point 2.1 to 2.7, with optional reference to for part 2.6) of this document.

The Developer needs to refer to points numbered 3 to 5 to get an overview for the appropriate system as required.

The Marketing staff can refer to the point 4, which specifies the specialized features of the said software.

Users can refer to the User Documentation provided at the point 2.6 of this document.

Testers need to test the software features mentioned in the point 4 and the nonfunctional requirements mentioned in the point 5

Product Scope

Windows 10 has been one of the biggest changes to Microsoft's operating system for decades. It features an all new design, a bunch of new features, a new personal assistant, a notification center, and a whole new platform. That platform, which perhaps marks the biggest change with Windows 10, is called the Universal Windows Platform, and it's a replacement for the old, so-called, Win32 platform that has been in use since 1993.

Microsoft is now focusing on UWP and the API expansion. Microsoft is now focusing mostly on UWP apps, and many people are not happy about that decision because they only see the present, not the future. The future is a clean and non-fragmented Windows, running on all your devices. On your *phone, gaming console, computer, laptop, tablet*, you name it.

It will most probably be called just "Windows." You won't have a traditional desktop PC in your room, but rather all you will have is a *HoloLens*. If you need to sit down and work in front of a desk, you just fire up Word and use your Bluetooth keyboard and mouse to do what you have to do. Think of it like Continuum for *Phones*, but a step further. You won't even need a screen anymore. You will have all your files in the cloud, and by using the glasses, you will just edit them wherever you go without begin limited to any device except your head.

References

https://www.onmsft.com/editorial/what-is-the-future-of-windows-10-and-what-will-happen-to-uwp-part-i

https://windows10-guide.com/

https://www.groovypost.com/howto/windows-10-edition-architecture-build-your-install/#:~:text=Determining%20Your%20Windows%2010%20Architecture,memory%20that%20can%20be%20installed.

https://techcommunity.microsoft.com/t5/ask-the-performance-team/windows-architecture-the-basics/ba-p/372345

https://download.microsoft.com/download/D/2/B/D2B18586-8C4F-4F40-828D-99D96489152A/Microsoft_Press_eBook_Introducing_Windows_10_Preview_PDF.pdf

Overall Description

Product Perspective

Windows 10 is a series of operating systems developed by Microsoft and released as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, released nearly two years earlier, and was released to manufacturing on July 15, 2015, and broadly released for the general public on July 29, 2015. Windows 10 was made available for download via MSDN and Technet, and as a free upgrade for retail copies of Windows 8 and Windows 8.1 users via the Windows Store. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10, which are available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

Product Functions

Windows 10, codenamed "*Threshold 1*", is the first release of Windows 10. It carries the build number 10.0.10240. While the build itself doesn't contain the version number, Microsoft retroactively named this version 1507, standing for July 2015 and matching the versioning scheme for later updates. "*Threshold 1*" was announced on an event on September 30, 2014 with a first preview following the day after. The final release was made available to Windows Insiders on July 15, 2015, followed by a public release on July 29, 2015 as a free upgrade to Windows 7 and Windows 8.1.

New functionalities indicated for this release are only those added since Windows 8.1 Update 1, released in April 2014. Following are the functionalities

o Bundled apps

o Microsoft Edge

o Development platform

o Gaming

o Shell and user interface

o Action Center

o Desktop

o Command line

o Cortana

o Lock screen

o Start menu

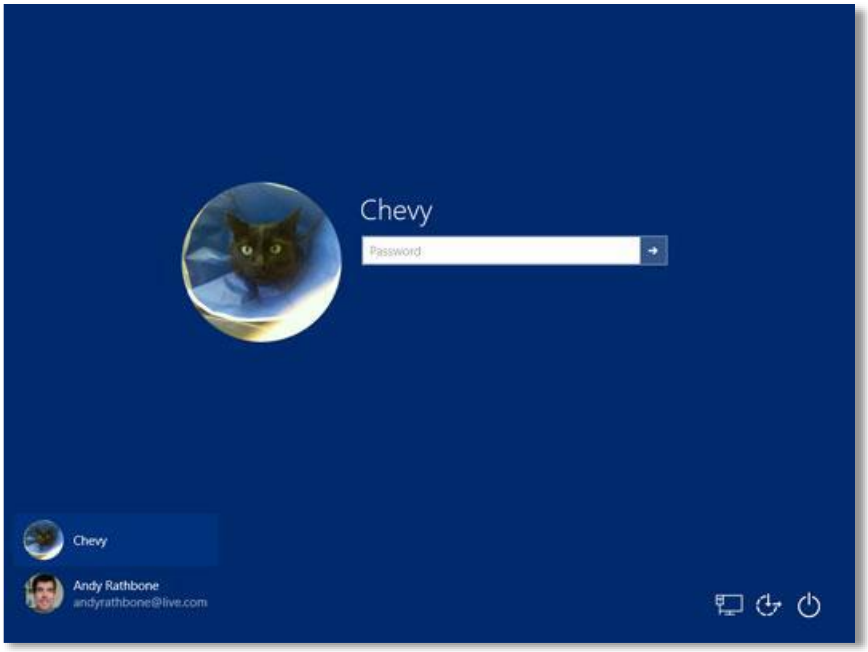
o Task View

o System settings

o Security

User Classes and Characteristics

Windows 10 wants you to set up a user account for everybody who uses your PC. A user account works like a cocktail-party name tag that helps Windows recognize who's sitting at the keyboard. Windows offers three types of user accounts: Administrator, Standard, and Guest. (It also offers a special Standard account for children.)



Windows gives each type of account permission to do different things on the computer. If the computer were a hotel, the Administrator account would belong to the desk clerk, each tenant would have a Standard account, and Guest accounts would belong to visitors trying to use the bathroom in the lobby. Here's how the different accounts translate into computer lingo:

Administrator: The administrator controls the entire computer, deciding who gets to play with it and what each user may do on it. On a computer running Windows, the owner usually holds the almighty Administrator account. He or she then sets up accounts for each household member and decides what they can and can't do with the PC. Standard: Standard account holders can access most of the computer, but they can't make any big changes to it. They can't run or install new programs, for example, but they can run existing programs.

Child: The Child account setting is actually just a Standard account with the Family Safety settings automatically turned on.

Guest: Guests can play with the computer, but the computer doesn't recognize them by name. Guest accounts function much like Standard accounts but with no privacy: Anybody can sign in with the Guest account, and the desktop will look the way the last guest left it. It's great for impromptu web browsing but not much else.

Operating Environment

The basic hardware requirements to install Windows 10 were initially the same as for Windows 8.1 and Windows 8, and only slightly higher than Windows 7 and Windows Vista. As of the May 2019 update, the minimum disk space requirement has been increased to 32 GB. In addition, on new installations, Windows permanently reserves up to 7 GB of disk space in order to ensure proper installation of future feature updates.

The 64-bit variants require a CPU that supports certain instructions. Devices with low storage capacity must provide a USB flash drive or SD card with sufficient storage for temporary files during upgrades.

Some pre-built devices may be described as "certified" by Microsoft. Certified tablets must include Power, Volume up, and Volume down keys; ⊞ Win and Rotation lock keys are no longer required.

Design and Implementation Constraints

The redesigned Start menu is the most obvious change, followed closely by the relocation of many system settings from Control Panel to the modern Settings app. ironically, the learning curve is considerably more complex if you and your users were early adopters of Windows 8.

Not only will you have to learn the new elements of Windows 10, but you’ll have to unlearn some of the techniques you mastered with Windows 8 and Windows 8.1. Feedback to Microsoft after the release of Windows 8 made it clear that the radically revised user experience caused significant frustration. Even with the refinements introduced in Windows 8.1, the change in user experience was substantial for anyone accustomed to the familiar desktop and Start menu.

As a result, the Windows 10 user experience offers another significant round of changes, designed to bring together the best elements of Windows 7 and Windows 8.1 and smooth the transition between the familiar desktop ways and the new touch-friendly techniques.

To install Windows 10, you need sufficient free storage space (at least 16 GB for 32-bit versions and 20 GB for 64-bit) and sufficient installed RAM (a minimum of 1 GB for 32-bit, 2 GB for 64-bit), or the installation will be blocked.

The processor must support Physical Address Extensions (PAE); Data Execution Protection, via the No-eXecute (NX) page-protection feature or the eXecute Disable (XD) bit feature; and Streaming SIMD Extensions 2 (SSE2).

A small number of older PCs might be blocked from 64-bit installations because their processors don’t support specific instructions like these: CMPXCHG16b, PrefetchW, and LAHF/SAHF.

The following device types are incompatible with Windows 10:

- The Surface RT, Surface 2, and other devices running Windows RT are not compatible with the Windows 10 Technical Preview and will not be upgradeable to the final release of Windows 10.

• Small tablets with 32 GB or less of storage that were configured using WIMBoot were blocked from upgrading to some releases of the Windows 10 Technical Preview. Microsoft has removed this limitation in current preview releases.

• The Windows 10 Mobile operating system, although closely related to Windows 10 in many respects, is delivered separately. The Windows 10 Technical Preview bits that are available for installation on PCs will not work on phones.

User Documentation

• Online video tutorials showing the uses of different new features and installation guide is there on YouTube.

• A PDF user manual to the user for better understanding and use is also provided on Microsoft official site.

• Many Website Showing different features and how to use them are also available.

Assumptions and Dependencies

Windows 10 also introduces two additional scenarios that organizations should consider:

- In-place upgrade, which provides a simple, automated process that leverages the Windows setup process to automatically upgrade from an earlier version of Windows. This process automatically migrates existing data, settings, drivers, and applications.
- Dynamic provisioning, which enables organizations to configure new Windows 10 devices for organization use without having to deploy a new custom organization image to the device.

Both of these scenarios eliminate the image creation process altogether, which can greatly simplify the deployment process.

Consider…	**For the Scenarios**
In-place upgrade	• When you want to keep all (or at least most) existing applications • When you do not plan to significantly change the device configuration (for example, BIOS to UEFI) or operating system configuration (for example, x86 to x64, language changes, Administrators to non-Administrators, Active Directory domain consolidations)

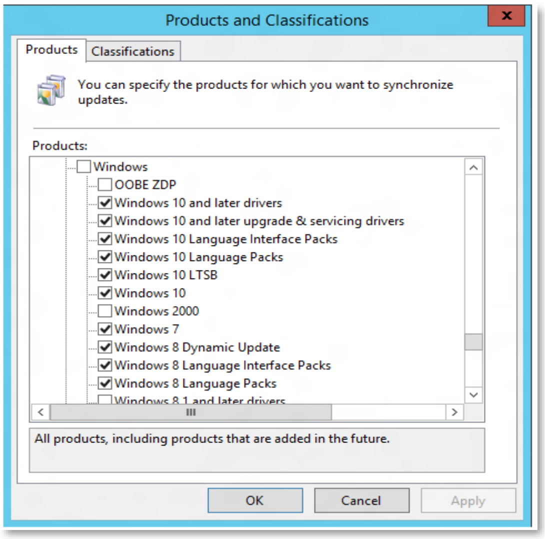
- To migrate from Windows 10 to a later Windows 10 release

Traditional wipe-and-load

- When you upgrade significant numbers of applications along with the new Windows OS
- When you make significant device or operating system configuration changes
- When you “start clean”. For example, scenarios where it is not necessary to preserve existing apps or data (for example, call centers) or when you move from unmanaged to well-managed PCs
- When you migrate from Windows Vista or other previous operating system versions

Dynamic provisioning

- For new devices, especially in “choose your own device” scenarios when simple configuration (not reimaging) is all that is required
- When used in combination with a management tool (for example, an MDM service like Microsoft Intune) that enables self-service installation of user-specific or role-specific apps



You can specify the products for which you want to synchronize updates.

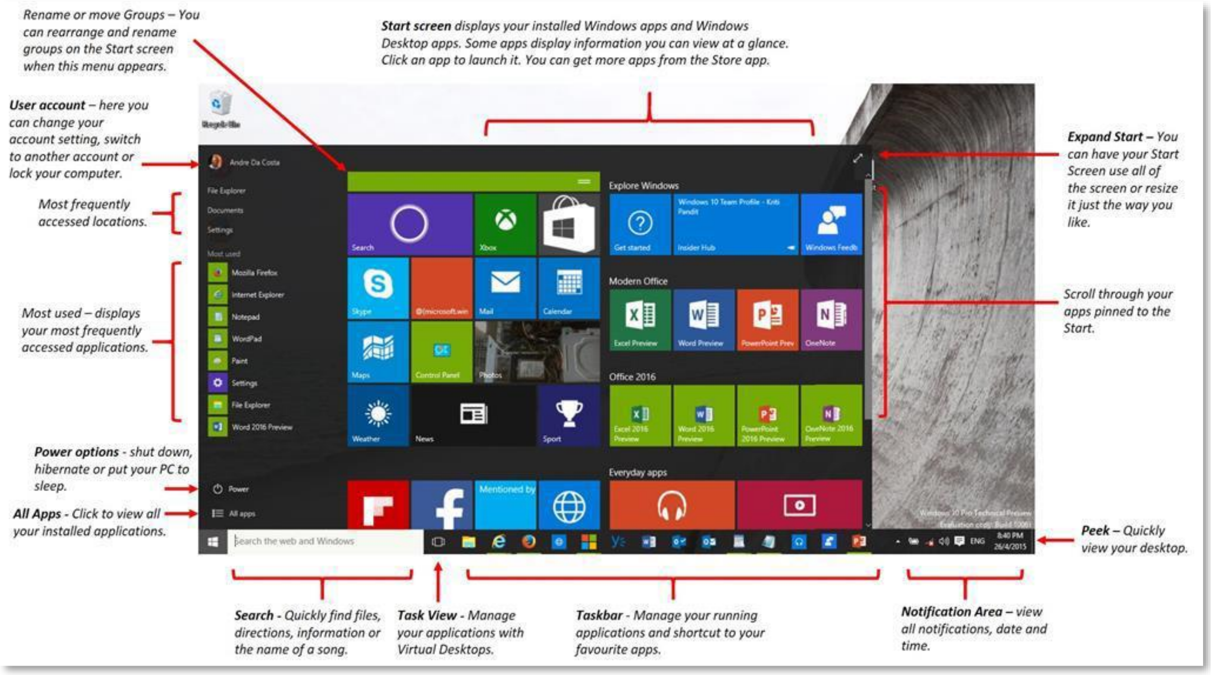
Products:

□ Windows
- □ OOBE ZDP
- ☑ Windows 10 and later drivers
- ☑ Windows 10 and later upgrade & servicing drivers
- ☑ Windows 10 Language Interface Packs
- ☑ Windows 10 Language Packs
- ☑ Windows 10 LTSB
- ☑ Windows 10
- □ Windows 2000
- ☑ Windows 7
- ☑ Windows 8 Dynamic Update
- ☑ Windows 8 Language Interface Packs
- ☑ Windows 8 Language Packs
- □ Windows 8.1 and later drivers

All products, including products that are added in the future.

External Interface Requirements

User Interfaces



Rename or move Groups – *You can rearrange and rename groups on the Start screen when this menu appears.*

Start screen *displays your installed Windows apps and Windows Desktop apps. Some apps display information you can view at a glance. Click an app to launch it. You can get more apps from the Store app.*

User account – *here you can change your account setting, switch to another account or lock your computer.*

Most frequently accessed locations.

Most used – *displays your most frequently accessed applications.*

Power options *- shut down, hibernate or put your PC to sleep.*

All Apps *- Click to view all your installed applications.*

Expand Start – *You can have your Start Screen use all of the screen or resize it just the way you like.*

Scroll through your apps pinned to the Start.

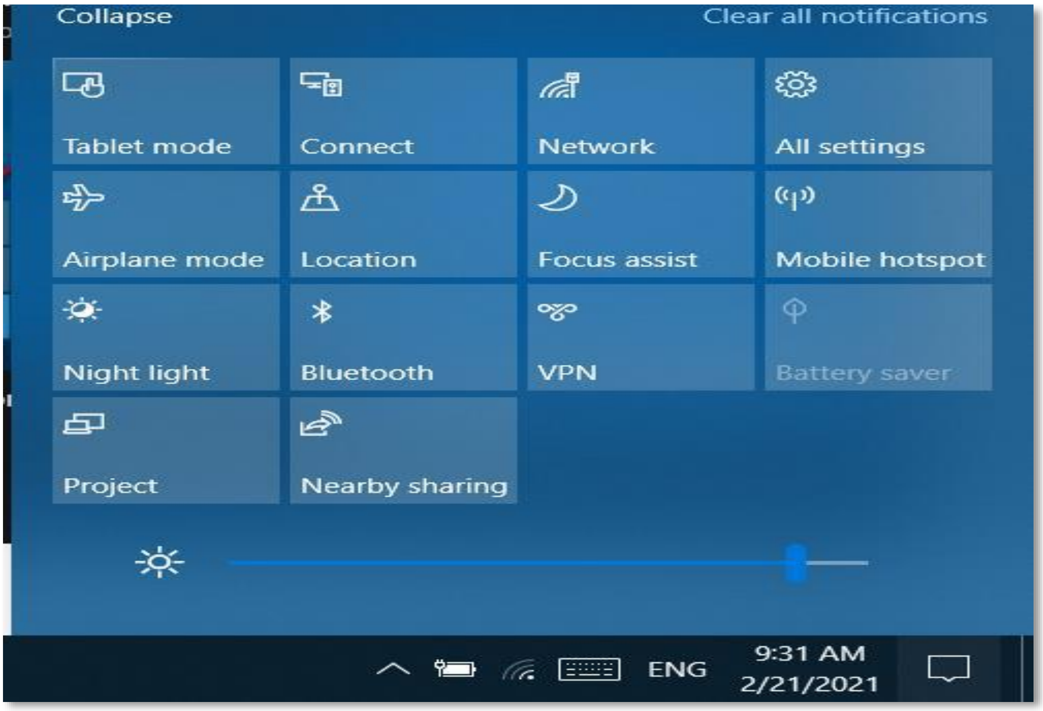
Peek – *Quickly view your desktop.*

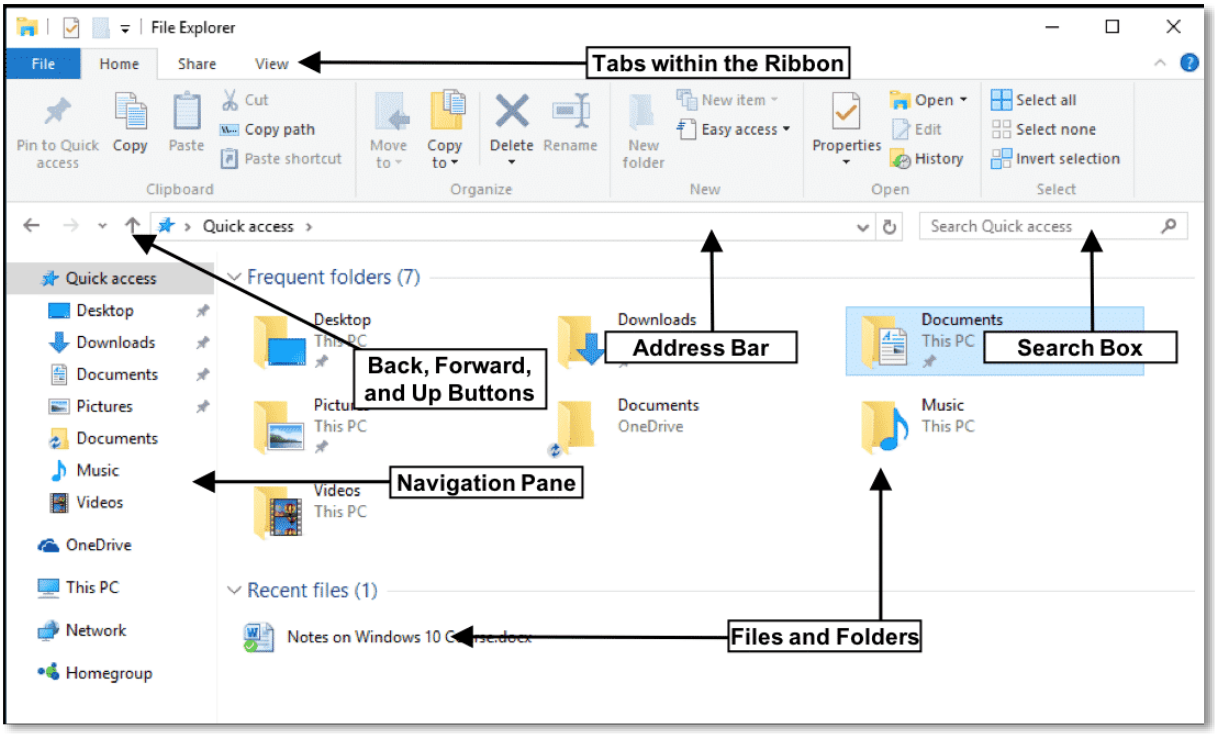
Search *- Quickly find files, directions, information or the name of a song.*

Task View *- Manage your applications with Virtual Desktops.*

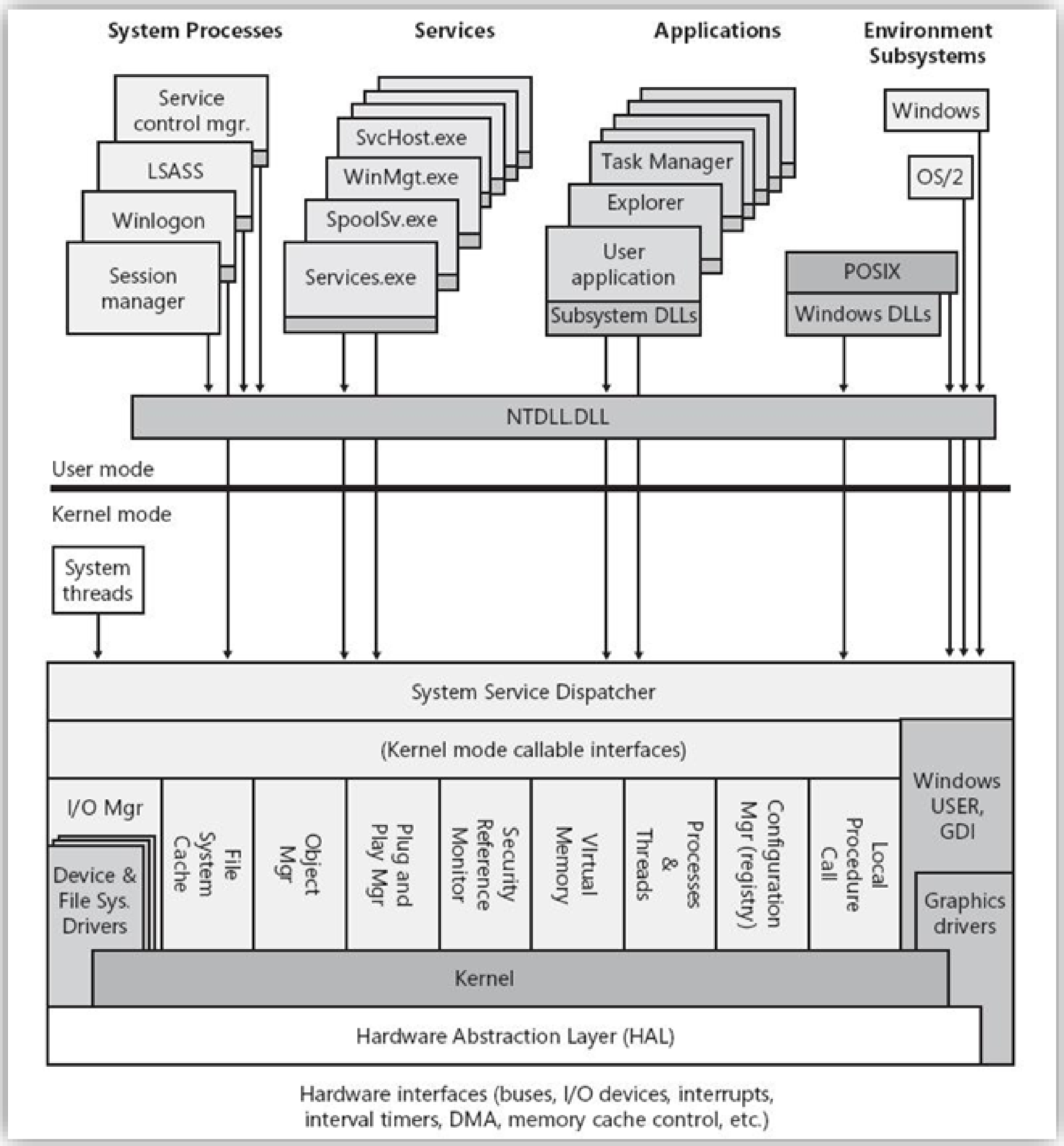
Taskbar *- Manage your running applications and shortcut to your favourite apps.*

Notification Area – *view all notifications, date and time.*

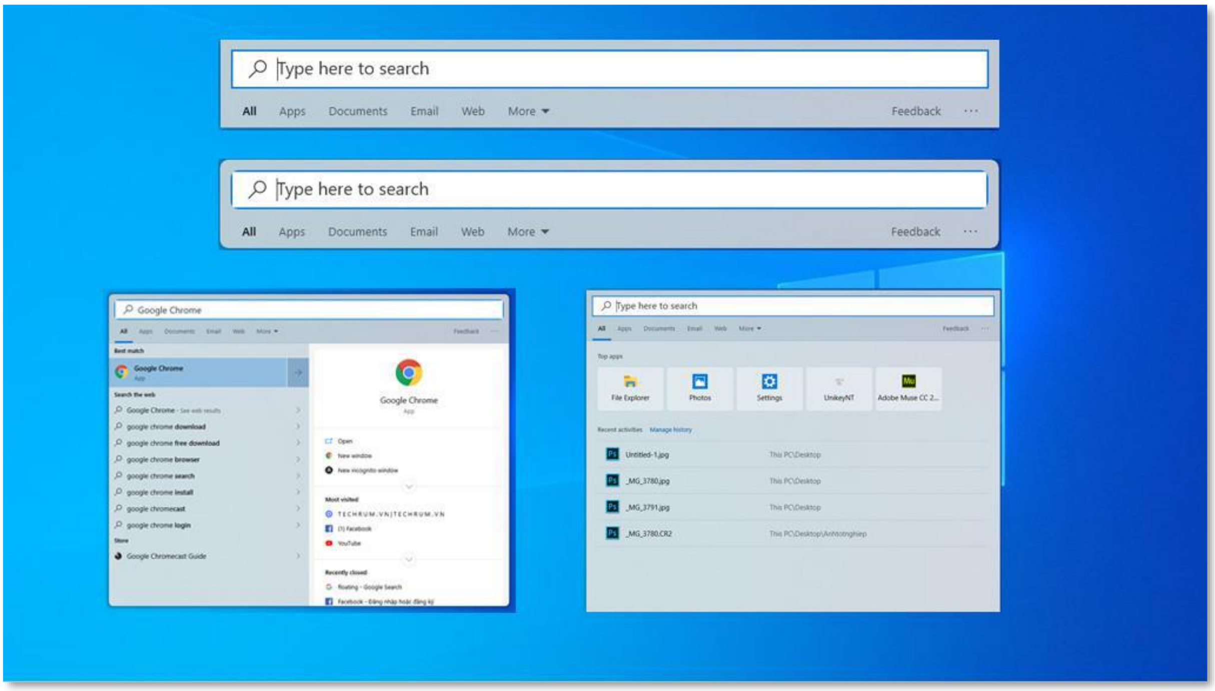


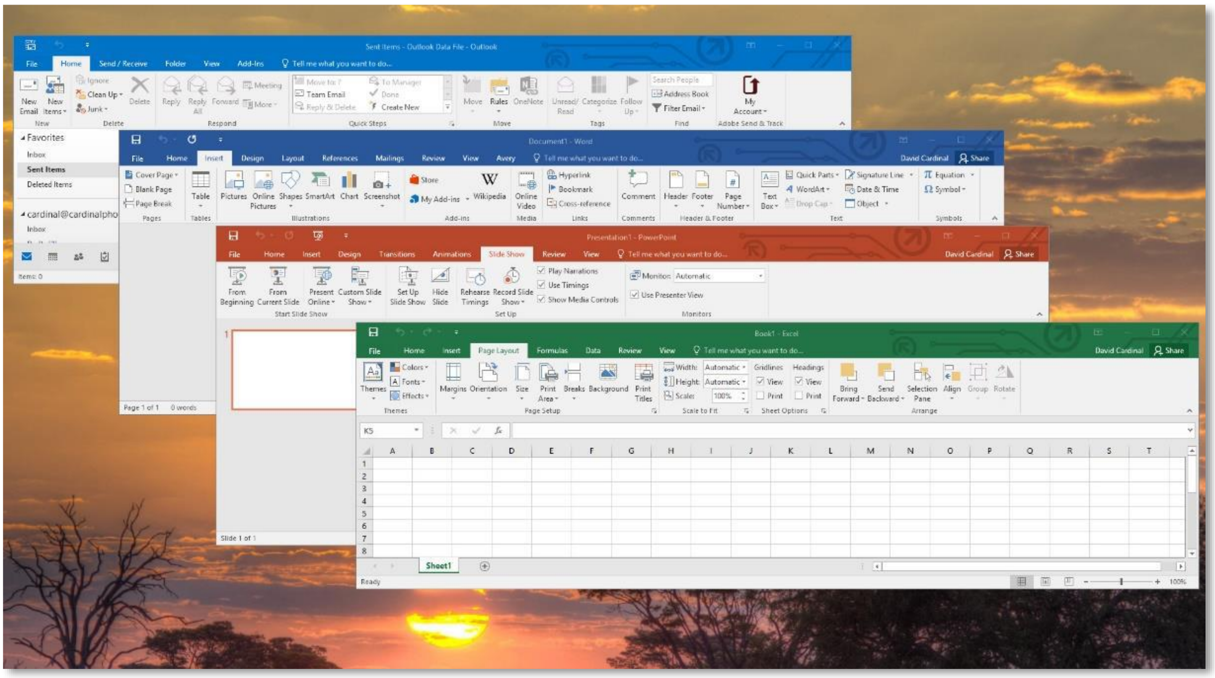

File Explorer
File
Home
Share
View
Tabs within the Ribbon
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organize
New folder
New item
Easy access
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
Quick access
Search Quick access
Address Bar
Search Box
Back, Forward, and Up Buttons
Navigation Pane
Files and Folders
Quick access
Desktop
Downloads
Documents
Pictures
Documents
Music
Videos
OneDrive
This PC
Network
Homegroup
Frequent folders (7)
Desktop
This PC
Downloads
Documents
This PC
Pictures
This PC
Documents
OneDrive
Music
This PC
Videos
This PC
Recent files (1)
Notes on Windows 10 Course.docx

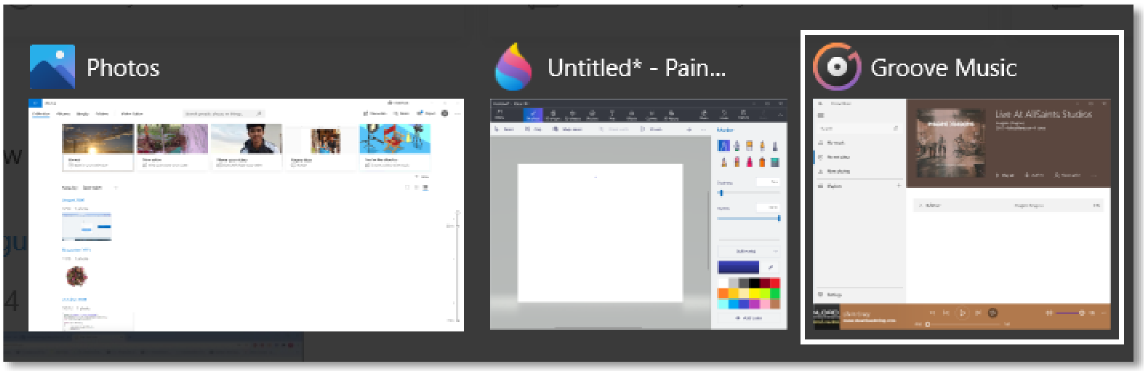
Hardware Interfaces



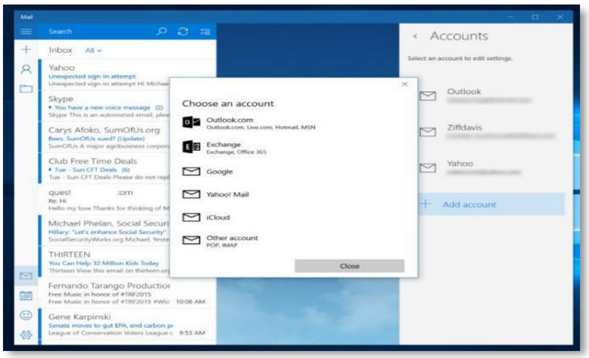
Software Interfaces

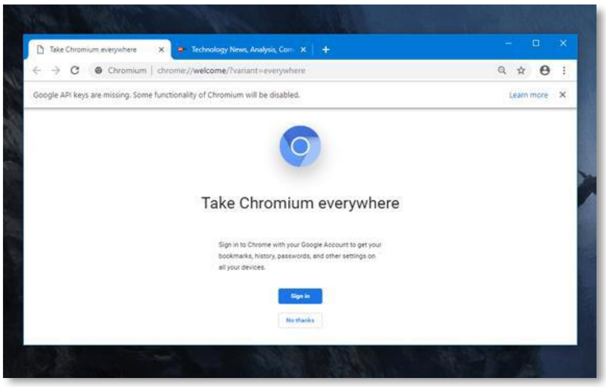






Communications Interfaces

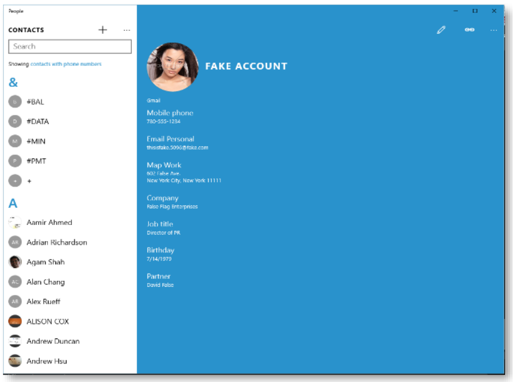


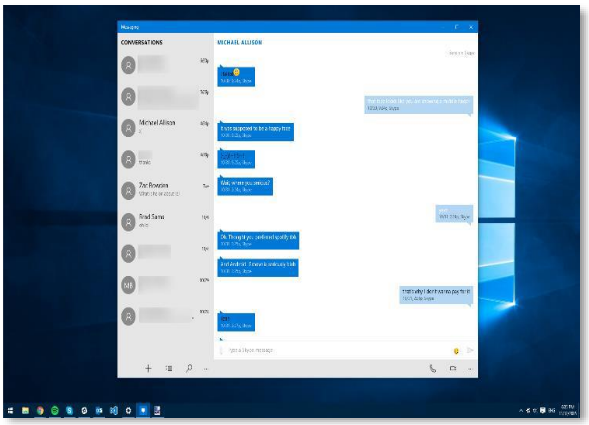

Take Chromium everywhere
Technology News, Analysis, Com
Chromium | chrome://welcome/?variant=everywhere
Learn more
Sign in
No thanks

Google API keys are missing. Some functionality of Chromium will be disabled.

Take Chromium everywhere

Sign in to Chrome with your Google Account to get your bookmarks, history, passwords, and other settings on all your devices.





System Features

4.1 Bundled apps

The Mail app adds user-configurable swipe gesture controls and POP3 email support. Google Calendar support is added to the ***Calendar app***.The Settings app is expanded to have similar functionality as the ***Control Panel,*** albeit with a Metro-style user interface.The ***Map app*** can download maps for offline use.

4.2 Microsoft Edge

Microsoft Edge is the new browser for Windows 10 and is the successor to Internet Explorer 11, although ***Internet Explorer*** will remain for compatibility and legacy purposes. ***Cortana*** has been integrated into Edge, accessible by the option "Ask Cortana" in the right click menu, as well as a Reading View and the ability to write notes directly on web pages and save to ***OneNote***. A Reading List feature has also been added, where users can save articles or other content to be accessed and read later. ***Microsoft Edge*** also includes a Share button on its toolbar where tapping or clicking on it will bring up the system Share panel, where users will be able to share a webpage to installed applications such as Reading List or third-party apps such as ***Facebook*** and ***Twitter***. Since its release, ***Microsoft Edge*** has scored 402 out of 555 points on HTML5test.

4.3 Development platform

Windows 10 introduced Universal Windows Platform (UWP), an extension of the Windows Runtime platform which was originally introduced with Windows 8. UWP emphasizes a core set of APIs common to all variations of the operating system, enabling the ability to code a single application with adaptations (such as user interface differences) for different device families and states, including *desktops* and *laptops, tablets, smartphones (via Windows 10 Mobile), Xbox One,* and other new device classes such as *Surface Hub* and *HoloLens.* An application may also react to the available displays and input on a device; when connected to a monitor or a suitable docking station, a UWP app on a smartphone can take on the appearance of the app on a PC. Information can also be synchronized between versions of an app for different devices, such as notifications and licensing.

4.4 Gaming

4.4.1 DirectX 12

Windows 10 includes DirectX 12 alongside WDDM 2.0. Unveiled March 2014 at GDC, DirectX 12 aims to provide "console-level efficiency" with "closer to the metal" access to hardware resources, and reduced CPU and graphics driver overhead. Most of the performance improvements are achieved through low-level programming, which can reduce single-threaded CPU bottlenecking caused by abstraction through higher level APIs. The performance gains achieved by allowing developers direct access to GPU resources is similar to other low-level rendering initiatives such as AMD's Mantle, Apple's

Metal API or the OpenGL successor, Vulkan. WDDM 2.0 introduces a new virtual memory management and allocation system to reduce workload on the kernel-mode driver.

4.4.2 Xbox One integration

Windows 10 brings more updates to the ***Xbox app*** introduced in Windows 8. Games from the Xb0ox One can be streamed to any Windows 10 device excluding smartphones.

4.4.3 Game bar and game DVR

Windows 10 introduces the game bar, which provides screenshot and video capture functionality for Windows games. Users can invoke the game bar, record gameplay, or take a screenshot using the appropriate keyboard shortcuts. Windows 10 can also continuously capture gameplays in the background; this allows the user to request that the last few users defined moments of gameplay be saved to the hard disk. This is useful if a user wants to save and/or share a moment of gameplay but did not think to explicitly record it beforehand.

4.5 Shell and user interface

Windows 10 also allows web apps and desktop software (using either Win32 or .NET Framework), to be packaged for distribution on ***Windows Store***. Desktop software distributed through ***Windows Store*** is packaged using the App-V system to allow sandboxing. Web apps are executed from remote servers, and have access to Windows functions such as notifications and camera access. As with Windows 8, locally packaged web apps can be written using HTML and WinJS.

4.6 Action Center

What was once called "Action Center" in Windows 7, Windows Server 2008 R2 and their successor is now called Security and Maintenance. The title of "Action Center" in Windows 10 is usurped by a sidebar that provides a list of received notifications and a group of "Quick actions" buttons for different settings areas. It is accessed by clicking the Notifications icon in the system tray, or swiping from the right of the screen on touchscreens.

4.7 Command line

Windows 10 brings improvements to the system's command-line interface. Unlike in previous versions of Windows NT, the Win32 console windows can now be resized without any restrictions. It can be made to cover the full screen by pressing the Alt+↵ Enter combination on keyboard. Microsoft also enabled the use of standard keyboard shortcuts, such as those for cut, copy, and paste, within the console. Word wrapping and keyboard shortcuts to move the caret, select and manipulate text have become available. Other features such as word wrap and transparency were also included. The user has the option to disable the new features and return to the legacy console if they wish.

4.8 Continuum

Continuum is the blanket title for a group of features on Windows 10 that are designed to enable smoother transitions between a default interface mode designed for use with a keyboard and mouse, and an interface designed for touchscreen environments, especially on hybrid devices such as laplets. Enabling "Tablet mode" switches the primary interface to a full screen version of the Start menu, and opens all applications in a maximized view by default. The taskbar is also modified, adding a Back button next to the Start button, and by default, hiding buttons for opened and pinned applications. Task View is used as the primary means of switching programs. Windows can prompt to switch between these two modes, or automatically do so, if certain events occur, such as plugging in a keyboard or mouse to a *tablet*, switching a laplet to its *laptop* state, or vice versa.

4.9 Cortana

Windows 10 has brought the ***Cortana assistant*** from Windows Phone 8.1 to Windows 10. By default, ***Cortana*** appears as a search pane on the taskbar, but can be changed into a button, like in tablet mode, and can be activated by voice using the command "Hey Cortana", when a user searches the Start menu, or when a user searches the ***Cortana*** search pane. With ***Cortana***, users can ask ***Cortana*** questions about the weather, calendar events, and other types of notifications, along with online information. ***Cortana*** currently requires a Microsoft Account to function.

4.10 Start Menu

Windows 10 reintroduced the start menu as seen in versions of Windows prior to 8. However, unlike these versions, the new start menu includes live tile features from Windows 8. It is possible to resize the Start menu and view recently added and most used applications. It can also be made full screen for *tablet* users or users that prefer a Windows 8-like experience. The right hand side of the Start menu can be used to pin tiles. The menu can contain a limited amount of columns, depending on the screen resolution. These columns can be divided in groups that can all have their own title. Every group is divided into 6 or 8 other columns, depending on the user's settings, to allow either 6 or 8 small sized tiles next to each other.

4.11 Task View

Task View is a task switching and virtual desktop system, accessible via the taskbar button, keyboard shortcut Windows+Tab ↹, or swiping from the left of a touchscreen. Activating Task View shows a zoomed display of all windows currently opened on a specific monitor; clicking on a window switches to it. Task View can also be displayed when a window is snapped to half the screen or three windows are snapped to fourths of the screen, prompting for a window to occupy the remainder of the screen. Task View also allows the creation of virtual workspaces; windows can be dragged into and out of these workspaces.

4.12 System Settings

The modern Settings app from Windows 8 continues to evolve in Windows 10, incorporating more system setting configuration functionality from the Windows ***Control Panel***. The ultimate goal is to make the Settings app feature complete, obviating the need for the ***Control Panel***.

The Push-button reset function has been changed to utilize files from the current Windows installation to rebuild the system rather than a separate recovery image. System updates carry over into the new installation and do not have to be re-downloaded. The separate "Refresh" option is removed; users are now given explicit choices within the Reset process to remove all personal files and applications, keep personal files but remove applications, or perform a full factory reset.

4.13 Security

MAC Address Randomization in WiFi has been introduced to try to prevent third parties from using the MAC address to track devices.

Other Non-functional Requirements

Performance Requirements

Following things are required for better performance:

o **Processor:** Intel 8th generation processors (Intel i3/i5/i7/i9-7x), Core M3-7xxx , Xeon E3-xxxx, and Xeon E5-xxxx processors, AMD 8th generation processors (A Series Ax-9xxx, E-Series Ex-9xxx, FX-9xxx) or ARM64 processors (Snapdragon SDM850 or later)

o **RAM:** 4 gigabyte (GB) for 32-bit or 16 GB for 64-bit

o **SSD/NVMe:** at least 128 GB for both 64-bit and 32-bit OS

o **Graphics card:** DirectX 9 or later

o **Display resolution:** 800 x 600, a minimum diagonal display size for the primary display of 7-inches or larger.

Safety Requirements

A device might not be able to receive updates if the device hardware is incompatible, if it lacks current drivers or sufficient available hard drive space, or if it’s otherwise outside of the Original Equipment Manufacturers (“OEM”) support period. Visit the Windows Lifecycle Fact Sheet or the Lifecycle FAQ for Windows products to learn more about the servicing timeline for each feature update.

Some of the disk space needed for installing updates is only temporarily required. Typically, ten days after installing an update, a disk cleanup will be automatically performed to delete copies of the older, unneeded Windows files and free up space again. Not all features in an update will work on all devices.

An internet connection is required to perform updates and Internet access (ISP) fees might apply.

If you need assistance installing an update, Windows 10 Update Assistant may be able to help.

Security Requirements

▪ Unified Extensible Firmware Interface (UEFI) After 30 years, the PC BIOS has finally been retired. Its replacement is UEFI, a firmware interface that takes over the functions traditionally performed by the BIOS. UEFI plays a critical role in security with Windows 10, offering the Secure Boot capability and support for self-encrypted drives, for example. (I'll say more about both of those features later in this chapter.) UEFI has been a requirement for original equipment manufacturers (OEMs) to certify a system or hardware device for Windows 8 or later under the Windows Hardware Certification Program (formerly known as the Windows Logo program).

▪ Trusted Platform Module (TPM) A TPM is a hardware chip (sometimes included as part of another component, such as a network card) that supports high-grade encryption and prevents tampering with or unauthorized export of certificates and encryption keys. The TPM can perform cryptographic operations and store keys for BitLocker volumes and virtual smartcards. A TPM can also digitally sign data, using a private key that software can't access. The presence of a TPM enables several key features in Windows 10, including BitLocker drive encryption, Measured Boot, and Device Guard. I discuss all these features later in this chapter.

Windows Security is built-in to Windows 10 and includes an antirvirus program called Microsoft Defender Antivirus. (In previous versions of Windows 10, Windows Security is called Windows Defender Security Center). If you have another antivirus app installed and turned on, Microsoft Defender Antivirus will turn off automatically. If you uninstall the other app, Microsoft Defender Antivirus will turn back on automatically.

Business Rules

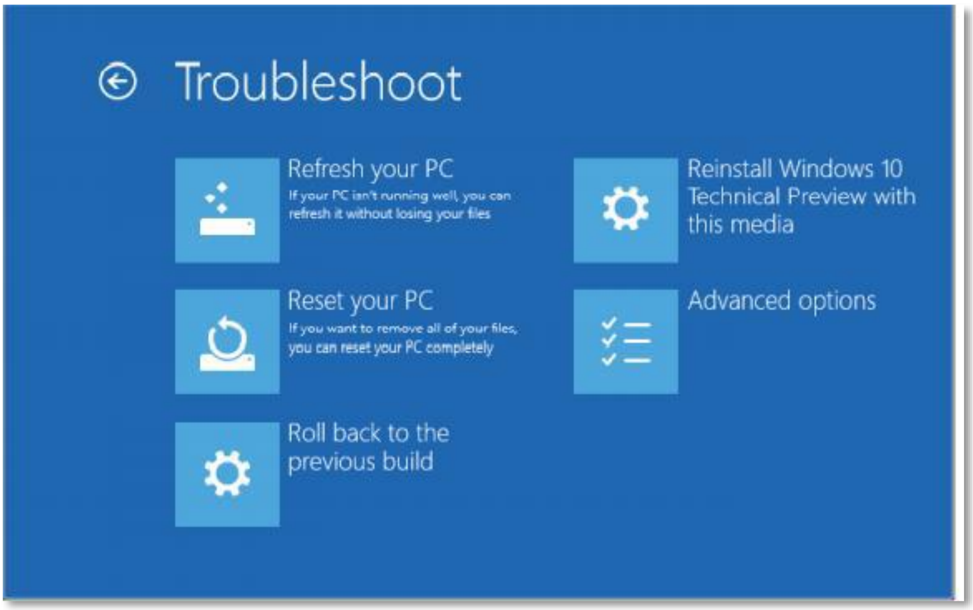
Enterprises running Windows 10 can develop universal line-of-business (LOB) apps and make them available to users inside their organization. These apps can be deployed in either of two ways: through a custom Business Store, managed and deployed by the Windows Store, or through a process called side loading.

In addition to creating and deploying apps, administrators can also use Group Policy to control the use of all apps, including those that are built in to Windows 10. For example, an organization might choose to remove the Sports app or prohibit it from running.The process of distributing a Windows 10 app through a private Business Store requires that an enterprise have Azure Active Directory accounts for each user in the organization. These accounts are used instead of Microsoft accounts. Installation files are managed and deployed by the Windows Store, which also tracks license usage. Updates are delivered via normal update channels—Windows Update or Windows Server Update Services (WSUS). LOB apps distributed within an organization without using the Windows Store don’t need to be signed by Microsoft, nor do they require Azure Active Directory accounts. They do, however, need to be signed with a certificate that is trusted by one of the trusted root authorities on the system. In this scenario, installation files are downloaded and deployed using the organization’s own infrastructure.

Other Requirements

Backup and recovery options in Windows 10

Windows 10 introduces major changes in the way the so-called “push-button reset” process works, eliminating the annoying problem of restoring an image that requires hours of updating before it’s useful and dramatically reducing the amount of space required as part of a standard install. For organizations that have a volume license agreement with Software Assurance, an additional, extremely powerful resource is available: the Microsoft Diagnostics and Recovery Toolset (DaRT).



Troubleshoot

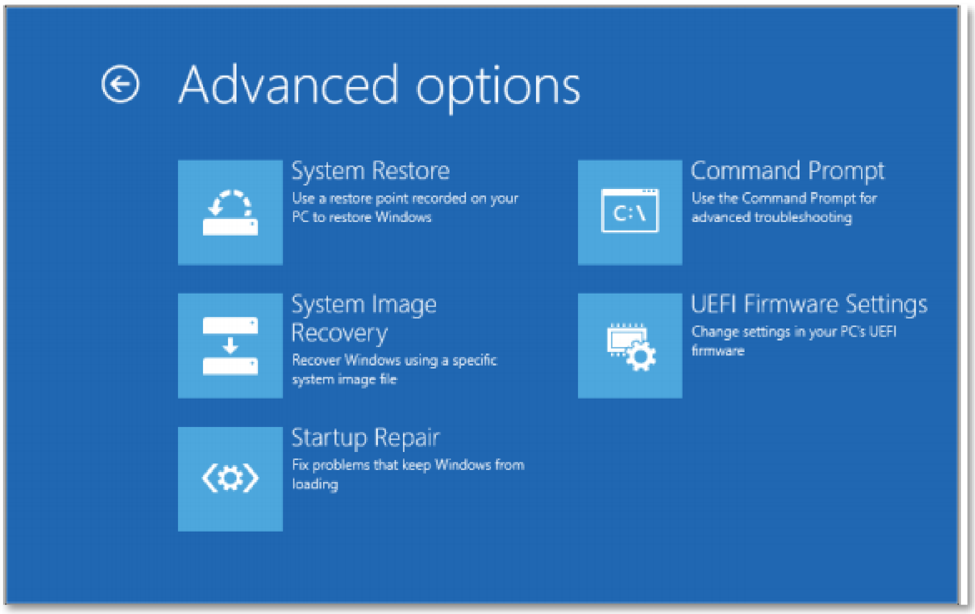
Refresh your PC
If your PC isn't running well, you can refresh it without losing your files

Reinstall Windows 10 Technical Preview with this media

Reset your PC
If you want to remove all of your files, you can reset your PC completely

Advanced options

Roll back to the previous build



Advanced options

System Restore
Use a restore point recorded on your PC to restore Windows

Command Prompt
Use the Command Prompt for advanced troubleshooting

System Image Recovery
Recover Windows using a specific system image file

UEFI Firmware Settings
Change settings in your PC's UEFI firmware

Startup Repair
Fix problems that keep Windows from loading

Appendix A: Glossary

✓ **About box**

A dialog box providing general program information, such as version identification, copyright, licensing agreements, and ways to access technical support.

✓ **access key**

An alphanumeric key that, when combined with the Alt key, activates a control. Access keys are indicated by underlining one of the characters in the control's label. For example, pressing Alt+O activates a control whose label is "Open" and whose assigned access key is "O". Access keys aren't case sensitive. The effect of activating a control depends on the type of control.

✓ **active monitor**

The monitor where the active program is running.

✓ **address bar**

A navigational element, usually appearing at the top of a window, that displays, and allows users to change, their current location.

✓ **application**

A program used to perform a related set of user tasks; often relatively complex and sophisticated.

✓ **application key**

A keyboard key with a context menu graphic on it. This key is used to display the context menu for the selected item.

✓ **application menu**

A control that presents a menu of commands that involve doing something to or with a document or workspace, such as file-related commands.

Displays the context menu for the current selection (the same as pressing Shift+F10).

✓ **application theming**

Using related visual techniques, such as customized controls, to create a unique look or branding for an application.

✓ **aspet ratio**

An expression of the relation between the width of an object and its height. For example, high definition television uses a 16:9 aspect ratio.

✓ **check box**

A common Windows control that allows users to decide between clearly differing choices, such as toggling an option on or off.

✓ **breadcrumb bar**

A navigational element, usually appearing at the top of a window, that displays, and allows users to change, their current location. "Breadcrumb" refers to breaking the current location into a series of links separated by arrows that users can interact with directly. Use address bar instead.

✓ **child window**

A window, such as a control or pane that is contained completely within another window referred to as the parent window.

✓ **constraint**

In controls that involve user input, such as text boxes, input constraints are a valuable way to prevent errors. For example, if the only valid input for a particular control is numeric, the control can use appropriate value constraints to enforce this requirement.

✓ **content area**

The portion of UI surfaces, such as dialog boxes, control panel items, and wizards, devoted to presenting options, providing information, and describing controls. Distinguished from the command area, task pane, and navigation area.

✓ **Control Panel**

A Windows program that collects and displays for users the system-level features of the computer, including hardware and software setup and configuration. From Control Panel, users can click individual items to configure system-level features and perform related tasks. See also: control panel item.

✓ **control panel item**

An individual feature available from Control Panel. For example, Programs and Ease of Access are two control panel items.

✓ **custom icon**

A pictorial representation unique to a program (as opposed to a Windows system icon).

✓ **default command button or link**

The command button or link that is invoked when users press the Enter key. The default command button or link is assigned by the developer, but any command button or link becomes the default when users tab to it.

✓ **default monitor**

The monitor with the Start menu, taskbar, and notification area.

✓ **desktop**

The onscreen work area provided by Windows, analogous to a physical desktop.

✓ **dialog box**

A secondary window that allows users to perform a command, asks users a question, or provides users with information or progress feedback.

✓ **dialog box launcher**

In a ribbon, a button at the bottom of some groups that opens a dialog box with features related to the group.

✓ **dialog unit**

A dialog unit (DLU) is the device-independent measure to use for layout based on the current system font.

✓ **direct manipulation**

Direct interaction between the user and the objects in the UI (such as icons, controls, and navigational elements). The mouse and touch are common methods of direct manipulation.

✓ **docked window**

A window that appears at a fixed location on the edge of its owner window.

✓ **drop-down arrow**

The arrow associated with drop-down lists, combo boxes, split buttons, and menu buttons, indicating that users can view the associated list by clicking the arrow.

✓ **drop-down list**

A common Windows control that allows users to select among a list of mutually exclusive values. Unlike a list box, this list of available choices is normally hidden.

✓ **effective resolution**

The physical resolution of a monitor normalized by the current dpi (dots per inch) setting. At 96 dpi, the effective resolution is the same as the physical resolution, but in other dpis, the effective resolution must be scaled proportionately. Generally, the effective resolution can be calculated using the following equation:

Effective resolution = Physical resolution x (96 / current dpi setting)

✓ **elevated administrator**

In User Account Control, elevated administrators have their administrator privileges. Without elevating, administrators run in their least-privileged state. The Consent UI dialog is used to elevate administrators to elevated status only when necessary.

✓ **error**

A state in which a problem has occurred.

✓ **floating window**

A window that can appear anywhere on the screen the user wants.

✓ **font**

A set of attributes for text characters.

✓ **full screen**

A maximized window that does not have a frame.

✓ **gadget**

A simple mini-application hosted on the user's desktop.

✓ **gallery**

A list of commands or options presented graphically. A results-based gallery illustrates the effect of the commands or options instead of the commands themselves. May be labeled or grouped. For example, formatting options can be presented in a thumbnail gallery.

✓ **gesture**

A quick movement of a finger or pen on a screen that the computer interprets as a command, rather than as a mouse movement, writing, or drawing.

✓ **getting started page**

An optional wizard page that outlines prerequisites for running the wizard successfully or explains the purpose of the wizard.

✓ **group box**

A common Windows control that shows relationships among a set of related controls.

✓ **handwriting recognition**

Software that converts ink to text.

✓ **Help**

User assistance of a more detailed nature than is available in the primary UI. Typically accessed from a menu or by clicking a Help link or icon, this content may take a variety of forms, including step-by-step procedures, conceptual text, or more visually-based, guided tutorials.

✓ **high-contrast mode**

A special display setting that provides extreme contrast for foreground and background visual elements (either black on white or white on black). Particularly helpful for accessibility.

✓ **landscape mode**

A presentation option that orients an object to be wider than it is tall.

✓ **indirect dialog box**

A dialog box displayed out of context, either as an indirect result of a task or as the result of a problem with a system or background process.

✓ **inductive user interface**

A UI that breaks a complex task down into simple, easily explained, clearly stated steps with a clear purpose.

✓ **ink**

The raw output for a pen. This digital ink can be kept just as written, or it can be converted to text using handwriting recognition software.

✓ **list view**

A common Windows control that allows users to view and interact with a collection of data objects, using either single selection or multiple selection.

✓ **live preview**

A preview technique that shows the effect of a command immediately on selection or hover without the user commiting the action. For example, formatting options such as themes, fonts, and colors benefit from live previews by showing users the effect with minimal effort.

✓ **localization**

The process of adapting software for different countries, languages, cultures, or markets.

✓ **log file**

A file-based repository for information of various kinds about activity on a computer system. Administrators often consult log files; ordinary users generally do not.

✓ **main instruction**

Prominently displayed text that concisely explains what to do in the window or page. The instruction should be a specific statement, imperative direction, or question. Good main instructions communicate the user's objective rather than focusing just on manipulating the UI.

✓ **managed environment**

A networked computer environment managed by an IT department or third-party provider, instead of by individual users. Administrators may optimize performance and apply operating system and application updates, among other tasks.

✓ **manipulation**

A type of touch interaction in which input corresponds directly to how the object being touched would react naturally to the action in the real world.

✓ **maximize**

To display a window at its largest size.

✓ **menu**

A list of commands or options available to users in the current context.

✓ **message box**

A secondary window that is displayed to inform a user about a particular condition

✓ **mini-toolbar**

A contextual toolbar displayed on hover.

✓ **minimize**

To hide a window.

✓ **mixed state**

For check boxes that apply to a group of items, a mixed state indicates that some of the items are selected and others are cleared.

✓ **multiple selection**

The ability for users to choose more than one object in a list or tree.

✓ **notification**

Information of a non-critical nature that is displayed briefly to the user; a notification takes the form of a balloon from an icon in the notification area of the taskbar.

✓ **opt in**

The ability for users to select optional features explicitly. Less intrusive to users than opt-out, especially for privacy and marketing related features, because there is no presumption of users' wishes. See also: opt out, options.

✓ **opt out**

The ability for users to remove features they don't want by clearing their selection. More intrusive to users than opt-in, especially for privacy and marketing related features, because there is an assumption of users' wishes. See also: opt in, options.

✓ **options**

Choices available to users for customizing a program. For example, an Options dialog box allows users to view and change program options.

✓ **owner window**

A window from which an owned window originates. Appears beneath the owned window in Z order.

✓ **page**

A basic unit of navigation for task-based UI, such as wizards, property sheets, control panel items, and Web sites. Users perform tasks by navigating from page to page within a single host window.

✓ **page flow**

A collection of pages in which users perform a task.

✓ **pane**

A rectangular area within a window that users may be able to move, resize, hide, or close. Panes are always docked to the side of their parent window. They can be adjacent to other panes, but they never overlap. Undocking a pane converts it to a child window.

✓ **parent window**

The container of child windows (such as controls or panes).

✓ **portrait mode**

A presentation option that orients an object to be taller than it is wide.

✓ **preview**

A representation of what users will see when they select an option. Previews can be displayed statically as part of the option, or upon request with a Preview or Apply button.

✓ **properties**

Settings of an object that users can change, such as a file's name and read-only status, as well as attributes of an object that users can't directly change, such as a file's size and creation date. Typically properties define the state, value, or appearance of an object.

✓ **protected administrator**

In User Account Control, an administrator running in their least-privileged state.

✓ **Quick Access Toolbar**

A small, customizable toolbar that displays frequently used commands.

✓ **Quick Launch bar**

A direct access point on the Windows desktop, located next to the Start button, populated with icons for programs of the user's choosing. Removed in Windows 7.

✓ **radio button**

A common Windows control that allow users to select from among a set of mutually exclusive, related choices.

✓ **restored window**

A visible, partial-screen window, neither maximized nor minimized.

✓ **scroll bar**

A control that allows users to scroll the content of a window, either vertically or horizontally.

✓ **secondary command**

A peripheral action that, while helpful, isn't essential to the purpose of the window. For example, Find Printer or Install Printer are secondary commands for a Print dialog box.

✓ **secondary window**

A window that has an owner window and consequently is not displayed on the taskbar.

✓ **secure desktop**

A protected environment that is isolated from programs running on the system, used to increase the security of highly secure tasks such as log on, password changes, and UAC

✓ **security shield**

A shield icon used for security branding.

✓ **settings**

Specific values that have been chosen (either by the user or by default) to configure a program or object.

✓ **shortcut key**

Keys or key combinations that users can press for quick access to actions they perform frequently. Ctrl+letter combinations and function keys (F1 through F12) are usually the best choices for shortcut keys. By definition, a shortcut key is the keyboard equivalent of functionality that is supported adequately elsewhere in the interface. Therefore, avoid using a shortcut key as the only way to access a particular operation.

✓ **Sidebar**

A region on the side of the user's desktop used to display gadgets in Windows Vista.

✓ **slider**

A common Windows control that displays and sets a value from a continuous range of possible values, such as brightness or volume.

✓ **splash screen**

Transitional screen image that appears as a program is in the process of launching.

✓ **supplemental toolbar**

A collection of commands designed to work in conjunction with a menu bar.

✓ **split button**

A bipartite command button that includes a small button with a downward pointing triangle on the rightmost portion of the main button. Users click the triangle to display variations of a command in a drop-down menu.

✓ **static text**

User interface text that is not part of an interactive control. Includes labels, main instructions, supplemental instructions, and supplemental explanations.

✓ **system menu**

A collection of basic window commands, such as move, size, maximize, minimize, and close, available from the program icon on the title bar, or by right-clicking a taskbar button.

✓ **task**

A unit of user activity, often represented by a single UI surface (such as a dialog box), or a sequence of pages (such as a wizard).

✓ **task dialog**

A dialog box implemented using the task dialog API. Requires Windows Vista or later.

✓ **task flow**

A sequence of pages that helps users perform a task, either in a wizard, explorer, or browser.

✓ **task link**

A link used to initiate a task, in contrast to links that navigate to other pages or windows, choose options, or display Help.

✓ **task pane**

A type of UI similar to a dialog box, except that it is presented within a window pane instead of a separate window. As a result, task panes have a more direct, contextual feel than dialog boxes. A task pane can contain a menu to provide the user with a small set of commands related to the selected object or program mode.

✓ **taskbar**

The access point for running programs that have a desktop presence. Users interact with controls called taskbar buttons to show, hide, and minimize program windows.

✓ **toolbar**

A graphical presentation of commands optimized for efficient access.

✓ **tooltip**

A small pop-up window that labels the unlabeled control being pointed to, such as unlabeled toolbar controls or command buttons.

✓ **touch**

Direct interaction with a computer display using a finger.

✓ **User Account Control**

With User Account Control (or UAC, formerly known as "Least-privilege User Account," or LUA) enabled, interactive administrators normally run with least user privileges, but they can self-elevate to perform administrative tasks by giving explicit consent with the Consent UI.

✓ **User Account Control shield**

A shield icon used to indicate that a command or option needs elevation for User Account Control.

✓ **user input problem**

An error resulting from user input. User input problems are usually non-critical because they must be corrected before proceeding.

✓ **user scenario**

A description of a user goal, problem, or task in a specific set of circumstances.

✓ **warning**

A message that describes a condition that might cause a problem in the future. Warnings aren't errors or questions. In Windows Vista and later, warning messages are typically displayed in task dialogs, include a clear, concise main instruction, and usually include a standard warning icon for visual reinforcement of the text.

✓ **welcome page**

The first page of a wizard, used to explain the purpose of the wizard. Welcome pages are no longer recommended. Users have a more efficient experience without such pages.

✓ **window**

A rectangular area on a computer screen in which programs and content appear. A window can be moved, resized, minimized, or closed; it can overlap other windows. Docking a child window converts it to a pane.

✓ **Windows logo key**

A modifier key with the Windows logo on it. This key is used for a number of Windows shortcuts, and is reserved for Windows use. For example, pressing the Windows logo key displays or hides the Windows Start menu.

✓ **wireframes**

A UI mockup that shows a window's functionality and layout, but not its finished appearance. A wireframe uses only line segments, controls, and text, without color, complex graphics, or the use of themes.

✓ **wizard**

A sequence of pages that guides users through a multi-step, infrequently performed task. Effective wizards reduce the knowledge required to perform the task compared to alternative UIs.

✓ **work area**

The onscreen area where users can perform their work, as well as store programs, documents, and their shortcuts.

Appendix C: To Be Determined List

https://www.microsoft.com/en-in/microsoft-365/windows

https://www.microsoft.com/en-in/windows/comprehensive-security

https://docs.microsoft.com/en-us/documentation/

https://techcommunity.microsoft.com/t5/ask-the-performance-team/windows-architecture-the-basics/ba-p/372345

https://download.microsoft.com/download/D/2/B/D2B18586-8C4F-4F40-828D-99D96489152A/Microsoft_Press_eBook_Introducing_Windows_10_Preview_PDF.pdf

CONCLUSION:

In this practical, we have learnt about Designing of Software Requirement Specification document on Microsoft Windows 10 Operating System in IEEE format.

PRACTICAL – 6

AIM: List at least 10 type of Testing for software development life cycle in IT industry and Design Test Case, Test Suites & Testing Strategy for the "Gmail" Web Application & Mobile Application.

THEORY

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Types of testing:

- Unit Testing
    - It focuses on the smallest unit of software design. In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs.
- Integration Testing
    - The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components is combined to produce output.
    - Integration testing is of four types: (i) Top-down (ii) Bottom-up (iii) Sandwich (iv) Big-Bang
- Regression Testing
    - Every time a new module is added leads to changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program.
- Smoke Testing
    - This test is done to make sure that software under testing is ready or stable for further testing
    - It is called a smoke test as the testing an initial pass is done to check if it did not catch the fire or smoke in the initial switch on.
- Alpha Testing
    - This is a type of validation testing. It is a type of acceptance testing which is done before the product is released to customers. It is typically done by QA people.

- Beta Testing
    - The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for a limited number of users for testing in a real-time environment.
- System Testing
    - This software is tested such that it works fine for the different operating systems. It is covered under the black box testing technique. In this, we just focus on the required input and output without focusing on internal working.
    - In this, we have security testing, recovery testing, stress testing, and performance testing.
- Stress Testing
    - In this, we give unfavourable conditions to the system and check how they perform in those conditions.
- Performance Testing
    - It is designed to test the run-time performance of software within the context of an integrated system.
    - It is used to test the speed and effectiveness of the program. It is also called load testing. In it we check, what is the performance of the system in the given load.
- Object-Oriented Testing
    - This testing is a combination of various testing techniques that help to verify and validate object-oriented software. This testing is done in the following manner:
        - Testing of Requirements,
        - Design and Analysis of Testing,
        - Testing of Code,
        - Integration testing,
        - System testing,
        - User Testing.
    - We use this OOT, for discussing test plans and for executing the projects.

TEST SUITE-1 FOR "GMAIL" Web Application & Mobile Application:

Test suite Summary:

For application such as GMAIL which is going to be used as email service, we have divided the test cases into two parts. This test suite will contain test cases related to Inbox functionality.

Test suite Design:

This test suite is designed for both Web application and mobile application. The pre-condition and post-conditions of web application and mobile application are listed together and expected output were merged with test cases table. Only one screenshot is presented in document section as the main screen of testing software.

Pre-conditions:

The preconditions to test the GMAIL is listed below.

- You need to have a web browser for web application.
    - Supported browsers for web application:
        - Google chrome
        - Firefox
        - Safari
        - Internet Explorer or Microsoft Edge
- For inbox functionality testing, you need to login to the GMAIL with valid credentials.
    - If you don't have credential, you can register yourself to get them.
- It is mandatory to have internet connection to connect your device with the server.
    - Good internet connection should yield faster result.

Post-conditions:

- The log-out functionality should work after the completion of testing.
- The backend should perform garbage collection after session completion.
- The connection with the database should be successfully released.

Risk assessment/Analysis:

Email risk assessment is the practice of validating a potential customer's email address against information about that person that exists online.

- Reverse Email Lookup:
    - Email background checks, also known as reverse email lookup, let you check personal information about customers by searching public databases and social networks for links.
- Sending malware using mail:
    - Email viruses, which constitute the majority of computer viruses, consists of malicious code that is distributed in email messages, and it can be activated when a user clicks on a link in an email message, downloads an email attachment or interacts in some other way with the body of an infected email.
- Sending suspicious URLs:
    - If the domain of a shortened URL is blacklisted, URLs are considered suspicious. So, avoid publicly available blacklisted URL shorteners. URLs should not be used to seek personal information such as email address, passwords, residential address, phone number, and credit card information.
- Overloading of spam:
    - Email spam, also referred to as junk email, is unsolicited messages sent in bulk by email (spamming).

TEST CASE	EXPECTED RESULT
Verify that on clicking ‘compose’ button, a frame to compose a mail gets displayed.	A compose screen should be visible.
Verify that user can enter email ids in ‘to’, ‘cc’ and ‘bcc’ sections and also user will get suggestions while typing the emailds based on the existing emailids in user’s email list.	Different input boxes should be displayed for ‘TO’, ‘CC’ and ‘BCC’ and while typing suggestions should be seen.
Verify that the user can enter multiple comma-separated emailids in ‘to’, ‘cc’ and ‘bcc’ sections.	Multiple inputs should be accepted.
Verify that the user can type subject line in the ‘subject’ textbox.	‘SUBJECT’ textbox should accept any text.
Verify that the user can type the email in the email-body section.	Body should be able to hold upto 384000 characters.
Verify that users can format mail using editor-options provided like choosing font-family, font-size, bold-italic-underline, etc.	Fonts should be customized.
Verify that the user can attach file as an attachement to the email.	The file should be shown as attachment.
Verify that the user can add images in the email and select the size for the same.	The custom image from user’s device or from URL should be attached.
Verify that sent mails can be found in ‘Sent Mail’ sections of the sender.	‘Sent mail’ should have sent mail.
Verify that mail can be sent to non-gmail emailIds also.	Mail should be sent to different domains too.
Verify that the emails composed but not sent remain in the draft section.	The ‘draft’ section should contain that mail.

TEST SUITE-2 FOR “GMAIL” Web Application & Mobile Application:

Test suite Summary:

For application such as GMAIL which is going to be used as email service, we have divided the test cases into two parts. This test suite will contain test cases related to Compose mail functionality.

Test suite Design:

This test suite is designed for both Web application and mobile application. The pre-condition and post-conditions of web application and mobile application are listed together and expected output were merged with test cases table. Only one screenshot is presented in document section as the main screen of testing software.

Pre-conditions:

The preconditions to test the GMAIL is listed below.

- You need to have a web browser for web application.
    - Supported browsers for web application:
        - Google chrome
        - Firefox
        - Safari
        - Internet Explorer or Microsoft Edge
- For inbox functionality testing, you need to login to the GMAIL with valid credentials.
    - If you don’t have credential, you can register yourself to get them.
- It is mandatory to have internet connection to connect your device with the server.
    - Good internet connection should yield faster result.

Post-conditions:

- The log-out functionality should work after the completion of testing.
- The backend should perform garbage collection after session completion.
- The connection with the database should be successfully released.

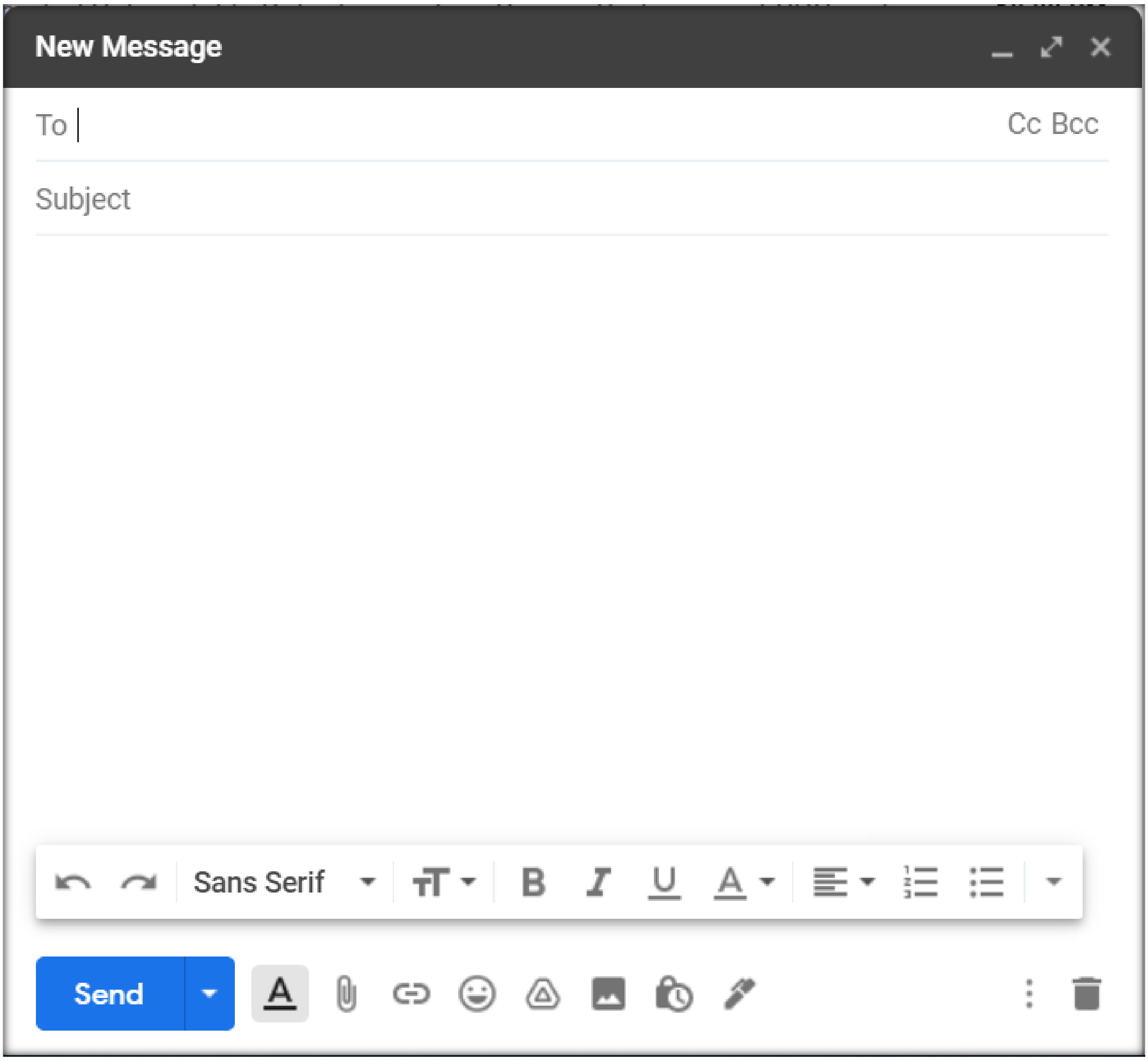
Risk assessment/Analysis:

Email risk assessment is the practice of validating a potential customer's email address against information about that person that exists online.

- Reverse Email Lookup:
    - Email background checks, also known as reverse email lookup, let you check personal information about customers by searching public databases and social networks for links.
- Sending malware using mail:
    - Email viruses, which constitute the majority of computer viruses, consists of malicious code that is distributed in email messages, and it can be activated when a user clicks on a link in an email message, downloads an email attachment or interacts in some other way with the body of an infected email.
- Sending suspicious URLs:
    - If the domain of a shortened URL is blacklisted, URLs are considered suspicious. So, avoid publicly available blacklisted URL shorteners. URLs should not be used to seek personal information such as email address, passwords, residential address, phone number, and credit card information.
- Overloading of spam:
    - Email spam, also referred to as junk email, is unsolicited messages sent in bulk by email (spamming).

TEST CASE	EXPECTED RESULT
Verify that on clicking ‘compose’ button, a frame to compose a mail gets displayed.	A compose screen should be visible.
Verify that user can enter email ids in ‘to’, ‘cc’ and ‘bcc’ sections and also user will get suggestions while typing the emailds based on the existing emailids in user’s email list.	Different input boxes should be displayed for ‘TO’, ‘CC’ and ‘BCC’ and while typing suggestions should be seen.
Verify that the user can enter multiple comma-separated emailids in ‘to’, ‘cc’ and ‘bcc’ sections.	Multiple inputs should be accepted.
Verify that the user can type subject line in the ‘subject’ textbox.	‘SUBJECT’ textbox should accept any text.
Verify that the user can type the email in the email-body section.	Body should be able to hold upto 384000 characters.
Verify that users can format mail using editor-options provided like choosing font-family, font-size, bold-italic-underline, etc.	Fonts should be customized.
Verify that the user can attach file as an attachement to the email.	The file should be shown as attachment.
Verify that the user can add images in the email and select the size for the same.	The custom image from user’s device or from URL should be attached.
Verify that sent mails can be found in ‘Sent Mail’ sections of the sender.	‘Sent mail’ should have sent mail.
Verify that mail can be sent to non-gmail emailIds also.	Mail should be sent to different domains too.
Verify that the emails composed but not sent remain in the draft section.	The ‘draft’ section should contain that mail.

Document:



TEST STRETEGY FOR “GMAIL” Web application & Mobile application

Scope:

The Gmail or Google Mail is a free email service introduced by Google. It allows sending and receiving mails over the Internet.

We can also send an email to multiple users at a time. The Gmail site is a type of Webmail.

We can access the Gmail from Web and as an application in Mobile devices. We can also use the third-party program to access the Gmail. Such programs synchronize the email content through the protocols IMAP (Internet Message Access Protocol) or POP (Post Office Protocol).

This document will be approved by both the testing manager and the development managers.

Test Approach:

For email service such as GMAIL whose target audience is in wide range, the testing should be done in multiple ways to ensure better security.

We propose to use Unit testing, Integration testing, Alpha testing and Beta testing to be used for current version.

UNIT TESTING is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

INTEGRATION TESTING is a level of software testing where individual units / components are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

But according to ISTQB Definition, integration testing is Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public. Alpha Testing is one of the user acceptance testing.

Beta Testing is performed by real users of the software application in a real environment. Beta testing is one of the type of User Acceptance Testing.

Test Environment:

A testing environment is a setup of software and hardware for the testing teams to execute test cases. In other words, it supports test execution with hardware, software and network configured. Test bed or test environment is configured as per the need of the Application Under Test.

For testing of Web application, we will use google Chrome browser with operating system either Windows 7 or higher, Linux or Mac.

For testing of Mobile application, we suggest to use android based operating system.

Testing Tools:

We are in an era of automation everywhere even in testing. So we will use some automated tools to perform testing. Though we will also use manual testing at places where it will be must.

We will use Selenium as testing tool.

Selenium is a testing framework to perform web application testing across various browsers and platforms like Windows, Mac, and Linux. Selenium helps the testers to write tests in various programming languages like Java, PHP, C#, Python, Groovy, Ruby, and Perl. It offers record and playback features to write tests without learning Selenium IDE.

Selenium proudly supports some of the largest, yet well-known browser vendors who make sure they have Selenium as a native part of their browser.

We will also use Watir.

Watir stands for Web Application Testing In Ruby. It facilitates the writing of automated tests by mimicking the behaviour of a user interacting with a website.

Watir is an open-source Ruby library for automating tests. Watir interacts with a browser the same way people do: clicking links, filling out forms and validating text.

Release Control:

This testing is defined for GMAIL latest web application and mobile application version 2021.02.21.361222438 which will be released in December 2019 and 9 March, 2021 respectively.

Risk Analysis:

Email risk assessment is the practice of validating a potential customer's email address against information about that person that exists online.

- Reverse Email Lookup:
  - Email background checks, also known as reverse email lookup, let you check personal information about customers by searching public databases and social networks for links.
- Sending malware using mail:
  - Email viruses, which constitute the majority of computer viruses, consists of malicious code that is distributed in email messages, and it can be activated when a user clicks on a link in an email message, downloads an email attachment or interacts in some other way with the body of an infected email.
- Sending suspicious URLs:
  - If the domain of a shortened URL is blacklisted, URLs are considered suspicious. So, avoid publicly available blacklisted URL shorteners. URLs should not be used to seek personal information such as email address, passwords, residential address, phone number, and credit card information.
- Overloading of spam:
  - Email spam, also referred to as junk email, is unsolicited messages sent in buk by email (spamming).

CONCLUSION:

In this practical, we learned about software testing. We learned about 10 widely used types of testing. We creates test cases and test suites for GMAIL which is one of the most famous email service around the world, we also wrote a test strategy for the same.

PRACTICAL – 7

AIM: Prepare the details Case Study on Design coding standards and guidelines for your respective SGP project definition and justify which Software Quality Standards & Testing Tool will be suitable for your SGP project.

THEORY

Different modules specified in the design document are coded in the Coding phase according to the module specification. The main goal of the coding phase is to code from the design document prepared after the design phase through a high-level language and then to unit test this code.

Purpose of Having Coding Standards:

- A coding standard gives a uniform appearance to the codes written by different engineers.
- It improves readability, and maintainability of the code and it reduces complexity also.
- It helps in code reuse and helps to detect error easily.
- It promotes sound programming practices and increases efficiency of the programmers.

Few coding standards include:

- Limited use of global
- Standard headers for different modules
- Naming conventions for local variables, global variables, constants and functions
- Proper indentation
- Error return values and exception handling conventions
- Avoid using a coding style that is too difficult to understand
- Avoid using an identifier for multiple purposes
- Code should be well documented
- Length of functions should not be very large
- Try not to use GOTO statement

Advantages of Coding Guidelines:

- Coding guidelines increase the efficiency of the software and reduces the development time.
- Coding guidelines help in detecting errors in the early phases, so it helps to reduce the extra cost incurred by the software project.
- If coding guidelines are maintained properly, then the software code increases readability and understand ability thus it reduces the complexity of the code.
- It reduces the hidden cost for developing the software.

SGP CASE STUDY

Introduction:

Our SGP is based on machine learning. We have created an easy-to-use mobile application to find whether a person is having COVID-19 or not from the X-ray image.

Programming language:

Our mobile application is made in React Native which is one of the famous mobile application development framework developed by Facebook.

Quality Standard:

- Functionality - We have tried to make it as user friendly as possible. We have added the functionalities such that user experience will increase.
- Reliability - We have tried to make it as reliable as possible. We have also used the best formulae to predict the probability for better understanding of the output.
- Usability - It is very easy to use so any user can easily access and use it.

Testing Tools:

Maveryx:

The Maveryx Test Automation Framework provides functional UI, regression, data-driven and keyword-driven testing capabilities for a wide range of desktop and Web technologies. Its innovative and intelligent technology inspects the application’s UI at runtime as a senior tester does. No code instrumentation, GUI capture, maps and object repositories. By Maveryx, you can start automation early and speed up your time-to-market without compromising on quality.

Selenium:

Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including C#, Groovy, Java, Perl, PHP, Python, Ruby and Scala. The tests can then run against most modern web browsers. Selenium runs on Windows, Linux, and macOS. It is open-source software released under the Apache License 2.0.

CONCLUSION

In this practical, we learned about coding standards and we used that concept on our SGP and derived some coding standard for it.

PRACTICAL – 8

AIM: List at least 5 online or offline Software Maintenance tools & prepare the detail case study on “Google Play Store” Maintenance document as a Google Play Store Developer.

THEORY

Software maintenance is a part of Software Development Life Cycle. Its main purpose is to modify and update software application after delivery to correct faults and to improve performance. Software is a model of the real world. When the real-world changes, the software requires alteration wherever possible.

The list of maintenance tools is given below.

- Bugzilla:
    - This program is used by thousands of software companies as a "defect tracking system."
    - The program keeps track of bugs that crop up in a software product and allows individuals or groups of developers to communicate and solve the problems.
    - Bugzilla allows software engineers to manage quality assurance, track bugs and submit bug-fixing patches to review.
    - The program is technically free, though they charge for extensive support issues.
    - Bugzilla is a secure program that has the ability to scan and correct database inconsistencies.

- HP Quality Center:
    - This Web-based application helps software engineers put their programs through the ringer by testing them for various bugs and errors.
    - HP Quality Center has the ability to plan and schedule tests, analyze results and manage issues and defects.
    - There are three versions of the program, the most extensive of which has the ability to deal with hundreds of applications at the same time with management teams distributed throughout the world.
    - The starter edition is available with a free download from the HP website.

- IBM Rational Quality Manager:
    - The IBM Rational Quality Manager is similar to the HP Quality Center in that it conducts tests using an Internet-based application.
    - The program lets users customize the way the information is presented through custom-designed dashboards, and it lets them schedule their software tests with the various members of their team.
    - These tests can be automated or done manually if there is a particular area of the software the engineers wish to get a closer look at.

  - The program description states that it will accelerate project schedules and ensure that applications meet the objectives of the company.
  - The express edition is available for download at the IBM site.Smoke Testing

- Micro Focus SilkPerformer:

  - Micro Focus SilkPerformer is a program that can be used by large enterprise-class companies to develop software applications through extensive testing.
  - The Micro Focus site states that the program can accelerate the resolution of problems by finding them early in development cycles.
  - SilkPerformer has full support for all Web 2.0 applications and efficient test cycles and test creation.
  - The program performs load tests that can be viewed across many environments in an effort to quickly determine if there are conditions that cause it to crash.
  - A trial version of the program is available from the Micro Focus site.

- Sage 300:

  - Provide companies with an overview of their business by assuring a single repository and compliance with procedures
  - Equip the technical services with a maintenance solution tailored to their business, easy to use and quick to deploy
  - Optimize spare parts, stocks and supplier’s management
  - Avoid the risk of errors between Sage 300 business management and CMMS
  - Apply the rules of business management to CMMS at every step of inventory and purchasing management process (e.g. replenishment threshold, control units, main supplier, …)

MAINTENANCE DOCUMENT FOR “GOOGLE PLAY STORE”

Objective:

Google play store is a platform where user can enjoy millions of the latest Android apps, games, music, movies, TV, books, magazines & more. The purpose of maintenance is to ensure the maximum efficiency and availability of production equipment, utilities and related facilities at optimal cost and under satisfactory conditions of quality, safety and protection for the environment.

Duties:

The maintenance team will take care of regular maintenance check. The regular check up of the system will be executed by them.

The maintenance manager will make sure that tasks are performed on time and he will report the updates.

Preventive Maintenance:

Google play store is a platform where users can upload their content so preventive maintenance plays important role.

The main activity of preventive maintenance is to keep a watch on the content that is being uploaded.

In case of upload of any harmful information, the strict actions will be taken against the user.

The routine check-up will involve the testing of all major components of software on monthly bases and solve detected bugs.

On yearly basis, User Interface should be checked for updates.

For optimizing the backend logic, half yearly check-up can be performed.

Corrective Maintenance:

The system failure will take the at most priority. The cause of crash should immediately be recognized and actions should be taken to fix it.

In case of malperformance of the system, the malfunctioning part can be isolated if known and it can be taken care of separately.

If malfunctioning part is not detected, the system can be out of service for certain amount of time and problem can be taken care of.

As the time passes, some functions or services can be deprecated, that would fall under maintenance as well. Those components should be replaced with another.

The Google play store will give every user the right to report bugs. Those bugs will be ordered by the danger for system and will be resolved altogether unless it poses potential damage to the system, such bugs are requested to solve as early as possible.

Maintenance execution and inspection:

The maintenance supervisor is the responsible for the correct execution of the maintenance work. When the work is done, he must register the shift remarks and positive variances in work order sheet if it was a scheduled corrective maintenance also evaluate de performance of the contractor services adding to the sheet or generating a work report as needed for futures references.

CONCLUSION:

In this practical, we learned about software maintenance. We found some tools for software maintenance. We wrote a brief case study on the maintenance document of one of the famous service provided by Google called Google Play Store.

PRACTICAL – 9

AIM: Prepare the detailed case study on Service-Oriented Architecture (SOA) of AWS services, Microsoft Azure services, Google Cloud Platform services as Cloud Computing Software Engineering developer in IT industry.

THEORY

Service-Oriented Architecture (SOA) is an architectural approach in which applications make use of services available in the network. In this architecture, services are provided to form applications, through a communication call over the internet.

- SOA allows users to combine a large number of facilities from existing services to form applications.
- SOA encompasses a set of design principles that structure system development and provide means for integrating components into a coherent and decentralized system.
- SOA based computing packages functionalities into a set of interoperable services, which can be integrated into different software systems belonging to separate business domains.

There are two major roles within Service-oriented Architecture:

- Service provider:
    - The service provider is the maintainer of the service and the organization that makes available one or more services for others to use. To advertise services, the provider can publish them in a registry, together with a service contract that specifies the nature of the service, how to use it, the requirements for the service, and the fees charged.
- Service consumer:
    - The service consumer can locate the service metadata in the registry and develop the required client components to bind and use the service.
    - Services might aggregate information and data retrieved from other services or create workflows of services to satisfy the request of a given service consumer. This practice is known as service orchestration Another important interaction pattern is service choreography, which is the coordinated interaction of services without a single point of control.

Advantages of SOA:

- Service reusability - In SOA, applications are made from existing services. Thus, services can be reused to make many applications.

- Easy maintenance - As services are independent of each other they can be updated and modified easily without affecting other services.
- Platform independent - SOA allows making a complex application by combining services picked from different sources, independent of the platform.
- Availability - SOA facilities are easily available to anyone on request.
- Reliability - SOA applications are more reliable because it is easy to debug small services rather than huge codes
- Scalability - Services can run on different servers within an environment, this increases scalability

Disadvantages of SOA:

- High overhead:
  - A validation of input parameters of services is done whenever services interact this decreases performance as it increases load and response time.
- High investment:
  - A huge initial investment is required for SOA.
- Complex service management:
  - When services interact, they exchange messages to tasks. the number of messages may go in millions. It becomes a cumbersome task to handle a large number of messages.

Practical applications of SOA:

SOA is used in many ways around us whether it is mentioned or not.

- SOA infrastructure is used by many armies and air force to deploy situational awareness systems.
- SOA is used to improve the healthcare delivery.
- Nowadays many apps are games and they use inbuilt functions to run. For example, an app might need GPS so it uses inbuilt GPS functions of the device. This is SOA in mobile solutions.
- SOA helps maintain museums a virtualized storage pool for their information and content.

CASE STUDY ON AWS

Introduction:

Service-Oriented Architecture (SOA) may be a kind of software design where services are provided to the opposite components by application components, through a communication protocol over a network. Its principles are independent of vendors and other technologies.

Implementations of SOA vary in terms of granularity: from only a few services that cover large areas of functionality to several dozens or many small applications in what's termed "microservice" architecture.

AWS Lambda:

AWS Lambda may be a service offered by the Amazon Web Services platform. AWS Lambda allows you to upload code that may be run on an on-demand container managed by Amazon. AWS Lambda will manage the provisioning and managing of servers to run the code, so all that's needed from the user may be a packaged set of code to run and some configuration options to define the context during which the server runs. These managed applications are mentioned as Lambda functions.

Modes of Operation:

AWS Lambda has 2 modes of operations.

- Asynchronous/Event-driven
  - Lambda functions is run in response to an occasion in asynchronous mode. Any source of events, such as S3, SNS, etc. won't block and Lambda functions can make the most of this in some ways, like establishing a processing pipeline for a few chains of events.
  - There are many sources of data, and counting on the source events are pushed to a Lambda function from the event source, or polled for events by AWS Lambda.
- Synchronous/Request->Response
  - For applications that need a response to be returned synchronously, Lambda are often run-in synchronous mode.
  - Typically, this is often utilized in conjunction with a service called API Gateway to return HTTP responses from AWS Lambda to an end-user, however Lambda functions may be called synchronously via a right away call to AWS Lambda.

AWS Lambda functions are uploaded as a zipper file containing handler code additionally to any dependencies required for the operation of the handler.

Lambda Functions as an Evolution of SOA:

Basic SOA could be a thanks to structure your code-base into small applications so as to learn an application within the ways described earlier during this article. Arising from this, the tactic of communication between these applications comes into focus. Event-driven SOA (aka SOA 2.0) allows for not only the normal direct service-to-service communication of SOA 1.0, but also for events to be propagated throughout the architecture so as to speak change.

Event-driven architecture may be a pattern that naturally promotes loose coupling and composability. By creating and reacting to events, services are often added ad-hoc to feature new functionality to an existing event, and several other events is composed to supply richer functionality.

CASE STUDY ON MICROSOFT AZURE

Introduction:

Azure is a new cloud computing platform under development by Microsoft (microsoft.com/windowsazure). Cloud computing allows developers to host applications in an Internet-accessible virtual environment. The environment transparently provides the hardware, software, network and storage needed by the application.

As with other cloud environments, Azure provides a hosted environment for applications. The added benefit of Azure is that .NET Framework applications can be deployed with minimal changes from their desktop siblings.

Applying service-oriented architecture (SOA) patterns and utilizing the experiences collected when implementing service-oriented solutions will be key to success when moving your services and applications into the new arena of cloud computing. To better understand how SOA patterns can be applied to Azure deployments, let's take a look at a scenario in which a fictional bank moves its services to the cloud.

Performance and Flexibility:

After some stress testing, the Woodgrove Bank development team found that having only one central data store in SQL Azure led to slower and slower response times when traffic increased. The developers decided to address this performance issue by using Azure table storage, which is designed to improve scalability by distributing the partitions across many storage nodes. Azure table storage also provides fast data access because the system monitors usage of the partitions and automatically load-balances them. However, because Azure table storage isn't a relational data store, the team had to design some new data storage structures and pick a combination of partition and row keys that would provide good response times.

Messaging and Queuing:

The objective is that no message should be lost even if services are offline due to error conditions or planned maintenance. The Asynchronous Queuing pattern allows this, though some offerings are not suitable for this pattern. For example, prompt answers with confirmation or denial of money transfers are necessary when dealing with online card transactions. But in another situation the pattern would do fine.

Communication between the Web and Worker roles is done with Azure Queues (as of the November CTP version it is possible to communicate directly between role instances), which are by default both asynchronous and reliable.

Putting Queues to Work:

As soon as a customer sends a message to UserAccountService, this message is placed in a Azure Queue and the customer receives a confirmation message. UserAccountWorker will then be able to get the message from the queue. Should UserAccountWorker be down, the message will not be lost as it is stored securely in the queue.

If the processing inside UserAccountWorker goes wrong, the message will not be removed from the queue. To ensure this, the call to the DeleteMessage method of the queue is made only after the work has been completed. If UserAccountWorker didn't finish processing the message before the timeout elapsed (the timeout is hardcoded to 20 seconds), the message will again be made visible on the queue so that another instance of UserAccountWorker can attempt to process it.

As soon as a customer sends a message to UserAccountService, this message is placed in a queue and the customer receives a confirmation message of type TransactionResponse. From the perspective of the customer, Asynchronous Queuing is used. ReliableMessaging is used to communicate between UserAccountStorageAction and AccountStorageWorker, which reside in the Web role and Worker role, respectively.

Processing Messages:

The ProcessMessage method first needs to get the content of the message. This can be done in one of two ways.

First, the message could be stored as a string in the queue.

Second, the message could be serialized XML.

Idempotent Capability:

What if one of Woodgrove Bank's customers sends a request to transfer money from one account to another and the message gets lost? If the customer resends the message, it is possible that two or more of the requests reach the services and gets treated separately.

One of the Woodgrove Bank team members immediately identified this scenario as one that requires the Idempotent Capability pattern. This pattern demands that capabilities or operations are implemented in such a way that they are safe to repeat. In short, the solution that Woodgrove Bank wants to implement requires well-behaved clients that attach a unique ID to each request and promise that they will resend the exact same message including the same unique ID in case of a retry. To be able to handle this, the unique ID is saved in the Azure table storage. Before processing any requests, it is necessary to check if a message with that ID was already processed. If it has been processed, a correct reply will be created, but the processing associated with the new request will not take place.

Although this means bothering the central data store with extra queries, it was deemed necessary. It will result in some deterioration of performance since some queries are made to the central data store before any other processing can take place. However, allowing this to consume extra time and other resources is a reasonable choice in order to meet Woodgrove Bank's requirements.

CONCLUSION:

In this practical, we learned about service-oriented architecture (SOA). We studied about its advantages and disadvantages. We learned about different platforms like AWS offered by Amazon, Azure offered by Microsoft and Google cloud offered by Google.

PRACTICAL – 10

AIM: Prepare the detailed case study as Software Development Document on MX Linux Operating System.

THEORY

Design Thinking:

Designers utilise design thinking to come up with new ideas and develop mobile apps. It is utilised to tackle complex problems and provide clients with desirable answers. It also aids in the resolution of unknown or ill-defined issues. Engineering, medical, business, architecture, sports, research, and mobile app development are just a few of the disciplines where information technology is applied.

Design thinking is a problem-solving strategy that is critical for improving user experience and understanding user demands. It's the ability to combine competing viewpoints to come up with fresh answers. It involves balancing the desirability, technical feasibility, and economic viability of a product or concept in terms of design. It also provides us with a fresh perspective on the issues. It is a human - centered, iterative design approach that typically includes five steps: problem definition, research, ideation, prototyping, and testing. These six steps, however, are not always in that order; they might occur simultaneously or repeatedly.

1. Research your target audience's demands and empathise with them to better understand them. Businesses must empathise with their customers in order to comprehend their demands and develop a solution.
2. Define – the issues based on the study findings, as well as your solutions to these issues.
3. Ideation entails questioning assumptions as well as brainstorming for new ideas and solutions. (If these concepts aren't viable, the ideas or solutions can be evaluated before prototyping begins.)
4. Prototype - to begin constructing a solution prototype.
5. Testing - the prototype to see how your ideas and solutions turn out, as well as their feasibility and viability.
6. After a prototype has been successfully tested, delivery and launch, implementation, iterating, and scaling take place.

Why is design thinking crucial in the development of mobile apps?

Innovation necessitates the use of design thinking. It is critical for a corporation to address an unmet client need through innovation and creativity. A company must constantly innovate and generate new, well-thought-out products in order to exist and succeed. This is when design thinking comes in handy. UI and UX are at the heart of every mobile app development, as all successful IT organisations understand. As a result, they must create apps that offer a distinct user experience.

Because consumers currently have easy access to worldwide marketplaces, they no longer distinguish between physical and digital experiences. As a result, it becomes increasingly difficult for businesses to differentiate their products or services from those of their competitors. Companies, on the other hand, can differentiate themselves from their competition by designing a mobile app that provides a distinctive user experience for end-users. As a result, IT organisations must work to bridge the gap between businesses and end-users.

According to studies, 80% of millennials carry their smartphones with them at all times of the day and night. They can't imagine life without their cell phones. It is now easy to use mobile applications due to the broad availability of the internet. Having a mobile app is a must these days. As a result, any company that wants to succeed and cater to millennials and other customers should develop mobile apps. Mobile apps have changed the face of tech firms and revolutionised the way they do business. Many marketers believe that using mobile apps is the most efficient approach to engage with clients and grow their brand. It's also simple to get straight feedback from customers.

Design thinking also lowers the risk of releasing innovative concepts. It promotes rapid learning and the development of exploratory, incremental, and novel ideas and solutions. When used appropriately, it also catches the attention of your target audience and aids in the generation of business.

- Because Design Thinking is based on a human-centred approach, it focuses on the end users and how to improve and enhance the user experience.
- Because it comprises diverse teams, the aggregate wisdom, expertise, and experience of all involved is available when developing solutions.
- It also entails coming up with unorthodox solutions. As a result, while solving genuine challenges, value is created and delivered to end-users.
- For design-driven companies, going to market with an MVP (minimum viable product) is ideal. In such circumstances, they take consumer feedback into account, incorporate it into their design, and produce a new version of the product. For example, Facebook, Instagram, WhatsApp, and other social media platforms.
- It aids in the creation of successful brands as well as the generation of revenue from these brands.

There are numerous advantages of developing a mobile app with unique design thinking:

1. Increased revenue — Because mobile applications are simple to use, they can reach a larger audience than traditional platforms. If the end-user experience is flawless, they will use your app again and again. This frequent use indicates brand loyalty, which might lead to an increase in sales.

2. Current and future marketing trends - Mobile apps are the need of the hour and the way of the future for a variety of apps and enterprises. As a result, developing mobile applications is critical for organisations.
3. Marketers may now focus on advertising their businesses via web and mobile platforms, thanks to the introduction of social media and mobile applications. The mobile apps can be accessed from anywhere in the world thanks to the internet's accessibility. Customer loyalty and brand reinforcement are also shown in the customer's use of the app on a regular basis.
4. Connecting with customers is simple – one of the biggest advantages of mobile apps is the ease with which you can connect with your customers. Customers can easily submit feedback, allowing businesses to learn from and fix their shortcomings straight from their customers.

CONCLUSION:

From this practical, I learn about the concept of design thinking. I also learn about the how design thinking methodology use in mobile application development.