Rapid App Development AT1

30003389 – Kyer Potts

Contents

[Q1. Provide a summary of the most common industry-accepted hardware and software that is used by Australian businesses. 1](#_Toc23823153)

[Hardware 1](#_Toc23823154)

[Software 1](#_Toc23823155)

[Network 2](#_Toc23823156)

[Storage 2](#_Toc23823157)

[Q2. Provide a definition and description of two major/popular operating systems used in the business world. 2](#_Toc23823158)

[Windows 2](#_Toc23823159)

[Linux 2](#_Toc23823160)

[Q3. Describe input and output drivers generally associated with Personal Computers used in a business environment. 3](#_Toc23823161)

[Q4.Provide a description of the major industry accepted prototyping tools. 3](#_Toc23823162)

[Q5. Provide a definition of Object-Oriented Programming and then a description of three major OOP languages. Give a code example of each using a binary search algorithm. 3](#_Toc23823163)

[C# (C Sharp) 3](#_Toc23823164)

[C# Binary Search Algorithm (GeeksforGeeks) 3](#_Toc23823165)

[Java 4](#_Toc23823166)

[Java Binary Search Algorithm (GeeksforGeeks) 5](#_Toc23823167)

[Python 5](#_Toc23823168)

[Python Binary Search Algorithm (GeeksforGeeks) 6](#_Toc23823169)

[Q6. Provide a definition of Procedural Programming and then a description of two major procedural programming languages. Give a code example of each using binary search algorithm. 6](#_Toc23823170)

[C 6](#_Toc23823171)

[C Binary Search Algorithm (GeeksforGeeks) 7](#_Toc23823172)

[Pascal 7](#_Toc23823173)

[Pascal Binary Search Algorithm (WisdomJobs) 8](#_Toc23823174)

[Q7. Provide a definition and description of real-time programming techniques. 10](#_Toc23823175)

[Q8. Provide a description of the software application measurement and estimation methodology outlined by the Consortium for IT Software Quality (CISQ) 10](#_Toc23823176)

[Software Sizing 10](#_Toc23823177)

[Code Quality 10](#_Toc23823178)

[Technical Debt 11](#_Toc23823179)

[Q9. Provide a description of software metrics used for software development. 11](#_Toc23823180)

[Product Metrics 11](#_Toc23823181)

[Process Metrics 11](#_Toc23823182)

[Project Metrics 11](#_Toc23823183)

[Q10. Provide a definition and description of three contemporary industry software development methodologies. Include a diagram of each SDLC. 12](#_Toc23823184)

[Waterfall Model 12](#_Toc23823185)

[Agile Model 12](#_Toc23823186)

[RAD Model 13](#_Toc23823187)

[Q11. Provide a definition of software development and configuration management processes. 14](#_Toc23823188)

[Core Patterns 14](#_Toc23823189)

[Workspace Patterns 14](#_Toc23823190)

[Codeline Patterns 14](#_Toc23823191)

[Q12. Provide a description of quality assurance practices and standards as they relate to Rapid Application Development that could be used by CITE Managed Services. 15](#_Toc23823192)

[Q13. Provide a description of Rapid Application Development and outline the role of the client, and their involvement in each phase of RAD. 15](#_Toc23823193)

[Q14. Describe the client business domain, and then outline the impact it will have on RAD; in particular the effect on the overall cost and quality of the final application. 16](#_Toc23823194)

[Q15. Create a comparison table of contemporary RAD tools, add the table headings like: targeted platform, RAD Type, cost, rating and major benefits. Your table should have more than 5 RAD tools. 16](#_Toc23823195)

[References 17](#_Toc23823196)

# Q1. Provide a summary of the most common industry-accepted hardware and software that is used by Australian businesses.

### Hardware

* **Computing Device:** Whether a laptop, or a desktop pc, a computing device allows its users to work productively in a modern office setting. Often, laptops will be the medium of choice for most office workers, as their portability allows user to work remotely, and move without greatly disrupting workflows or patterns. Desktop computers are traditionally reserved for power users, such as those in STEM research fields whom must manage and calculate high volumes of data, or media staff, who require high graphical processing power to handle high polygon counts or pixel rich graphics.
* **Seating Equipment:** Most offices will require seating arrangements for staff to work effectively, this can vary, as modes of seating can fall in and out of fashion, such as the swing between seated and standing desks, all the way to kneeling seats, couches, and even working hammocks that are employed by some organisations in an attempt to appear trendy to the public. Seating arrangements are often factored into the layout plan, and different arrangements can have pros and cons for office productivity.
* **Communication Equipment:** This can include desk phones, cell phones, radio equipment, internet capable devices, intercoms, routers and internet transmitting devices or anything else that can aid the business in achieving its communication goals.
* **Office Appliances**: Many of the traditional office appliances have fallen out of favour in recent years as industries continue to push towards paper free environments, however there are usually some uses still to be found for printers, copiers, shredders and even fax machines.
* **Peripherals and Accoutrements**: Things such as keyboards, mouses, monitors and standard office supplies fall under this category, many of things can be highly useful in an office environment, however the omission of them may not cause any issues depending on the availability of the other equipment on this list.

### Software

* **Office Software:** This includes applications that handle web processing, spreadsheets, presentations and slideshows, email and calendar management, desktop publishers and personal organisers and not taking software. The most popular of these applications is Microsoft Office suite which collects all the basic office applications Microsoft offer into a single self-contained package.
* **Internet Browsers:** Usually internet browsers will be offered at the user’s choice, generally between google chrome and Mozilla Firefox. Internet explorer has lagged behind these two browser powerhouses and can never seem to catch up.
* **Database Management Software:** Depending on the software system solution implemented for the office, database management software will be required. In a paperless office, the need for this software is doubly so. There are many applications available for database management, and many offer complete data solutions.
* **Project Management Software:** Depending on the needs and domain of the business, some form of project management tool may be required, the most popular of these being Trello for project planning and communication, and GitHub as a source control and distribution measure. Other software for detailed project planning such as MS Project is also available for purchase as a part of the Microsoft Office Suite package.
* **Domain Specific Software:** This choice of software depends on the operational needs of the business. Does the business operate within media? If so, media specific applications such as Adobe Suite might be required. Math and formula editors will be needed for research led businesses, consultancies will require diagram and brainstorming applications. This field of software has inexhaustible solutions for inexhaustible problems, the list goes on.

### Network

* **Routing Solution:** Provides a splitting option to provide internet to users within the office.
* **Switches:** Switches connect multiple devices to a network within a singular location.
* **Firewall:** Prevents malicious attacks from external entities and blocks unauthorised access to the network.
* **Wireless Access Points:** Provides access to the network for wireless devices.

### Storage

* **Cloud Storage:** Cloud storage saves and manages business data in an external location and is accessed via an online client. Cloud storage can provide several benefits, such as flexibility of access, disaster recovery and pricing structure, however there is an increased security risk associated with cloud storage solutions.
* **Server Based Storage (Hyper-Convergence):** This storage solution usually requires a server cluster stored within a data centre, usually located on business grounds. This solution provides safe and secure access to data from the intranet, at high speeds due to the proximity of the servers. It is however ungainly, and a server-based storage system will usually require a large amount of space and regular maintenance.
* **Traditional Storage Systems:** Many small businesses will operate with either Network Attached Storage (NAS) or Storage Area Network (SAN) Systems. Many users are familiar with these systems and they are scalable, provide fast access to data, are affordable in smaller scales and can be permission protected.

# Q2. Provide a definition and description of two major/popular operating systems used in the business world.

### Windows

Microsoft Windows is a home and business computing operating system with a graphical user interface aimed at ease of use for consumers. Windows was originally released in 1985 and has released many iterations over the following 30 years, most currently, Windows 10. Windows is currently the most popular operating system for laptop and desktop computers, and is installed on 82% of desktop computers worldwide

### Linux

Linux is an open source operating system created by Linus Torvalds in 1991. Linux is a GUI operating system based on Unix and has spawned many different operating systems that operate using the Linux Kernel such as Ubuntu and Mint. Most Linux based distributions are free and open source and are widely used by tech professionals for the power that the Linux command line offers to users.

# Q3. Describe input and output drivers generally associated with Personal Computers used in a business environment.

Computer attachments and peripherals and even some of the hardware itself will require drivers to operate effectively within the machine. Drivers are programs used to interface to hardware attached to the machine and enable software applications and operating systems to access the hardware via a bus port or communications subsystem, the driver then relays and translates information through the port so that the device can operate effectively.

# Q4.Provide a description of the major industry accepted prototyping tools.

Prototyping tools are used to streamline product creation and make the process of design and build more effective and robust. Many clients will request access to an interactive prototype of the project during certain intervals, and depending on the SLDC implemented, a prototype showcasing the new features generated by the current iteration of the product will need to be supplied. Prototyping tools can greatly aid in this process and there are many available. 3 of the most popular are listed below.

* **Proto.io:** Proto.io is a web-based tool with a large package of add-ons and features to aid in the prototyping process. It was originally created for mobile applications and follows an ease of use ethic with drag and drop features, prototype sharing and custom animation creation.
* **Adobe XD CC 2018:** Adobe XD implements a dual canvas design that streamlines the transition from design to prototype, which allows you to draw relationships via clickable elements and pages. XD CC is mainly used for mobile and web applications and is geared towards visual project prototypes.
* **Axure:** Axure is less visually based than Proto.io and Adobe XD CC and tends to work with complex projects requiring dynamic data. This technical focus allows wireframing of detail driven applications and is very useful for planning out larger scale projects. Axure also handles dynamic content extremely well compared to the previous two applications, using conditional logic and manipulating objects with variables and parameters.

# Q5. Provide a definition of Object-Oriented Programming and then a description of three major OOP languages. Give a code example of each using a binary search algorithm.

### C# (C Sharp)

C# is a general-purpose OOP (Object-Oriented Programming) language developed by a Microsoft team led by Anders Hejlsberg. C# is very similar to Java syntactically, as it shares the same roots in C and C++. C# operates within the .NET framework provided by Microsoft.

### C# Binary Search Algorithm (GeeksforGeeks)

// C# implementation of recursive Binary Search

using System;

class GFG {

    // Returns index of x if it is present in

    // arr[l..r], else return -1

    static int binarySearch(int[] arr, int l,

                            int r, int x)

    {

        if (r >= l) {

            int mid = l + (r - l) / 2;

            // If the element is present at the

            // middle itself

            if (arr[mid] == x)

                return mid;

            // If element is smaller than mid, then

            // it can only be present in left subarray

            if (arr[mid] > x)

                return binarySearch(arr, l, mid - 1, x);

            // Else the element can only be present

            // in right subarray

            return binarySearch(arr, mid + 1, r, x);

        }

        // We reach here when element is not present

        // in array

        return -1;

    }

    // Driver method to test above

    public static void Main()

    {

        int[] arr = { 2, 3, 4, 10, 40 };

        int n = arr.Length;

        int x = 10;

        int result = binarySearch(arr, 0, n - 1, x);

        if (result == -1)

            Console.WriteLine("Element not present");

        else

            Console.WriteLine("Element found at index "

                              + result);

    }

}

// This code is contributed by Sam007.

### Java

Java in an OOP developed by Sun Microsystems with the design philosophy that the language should be able to run on every device. This philosophy led to the unique compilation design that first compiles into byte code and is then run on Java Virtual Machine. This allowed java to be compiled and run on all platforms that have java supporting software and runtime environments and is not confined to a specific operating system, unlike C# which will only compile and run on .NET architecture.

### Java Binary Search Algorithm (GeeksforGeeks)

// Java implementation of recursive Binary Search

class BinarySearch {

    // Returns index of x if it is present in arr[l..

    // r], else return -1

    int binarySearch(int arr[], int l, int r, int x)

    {

        if (r >= l) {

            int mid = l + (r - l) / 2;

            // If the element is present at the

            // middle itself

            if (arr[mid] == x)

                return mid;

            // If element is smaller than mid, then

            // it can only be present in left subarray

            if (arr[mid] > x)

                return binarySearch(arr, l, mid - 1, x);

            // Else the element can only be present

            // in right subarray

            return binarySearch(arr, mid + 1, r, x);

        }

        // We reach here when element is not present

        // in array

        return -1;

    }

    // Driver method to test above

    public static void main(String args[])

    {

        BinarySearch ob = new BinarySearch();

        int arr[] = { 2, 3, 4, 10, 40 };

        int n = arr.length;

        int x = 10;

        int result = ob.binarySearch(arr, 0, n - 1, x);

        if (result == -1)

            System.out.println("Element not present");

        else

            System.out.println("Element found at index " + result);

    }

}

/\* This code is contributed by Rajat Mishra \*/

### Python

Python is a high level dynamically typed object-oriented programming language designed to be as close to normal speech as possible within the context of general purpose programming. It was developed by Guido van Rossum in 1991 and draws inspiration from C++, Java, Perl and Lisp. It is however an interpreted language, and does not need to be compiled to run. This has various advantages over compiled languages, however it is generally slower as it does not directly run machine code.

### Python Binary Search Algorithm (GeeksforGeeks)

# Python Program for recursive binary search.

# Returns index of x in arr if present, else -1

def binarySearch (arr, l, r, x):

    # Check base case

    if r >= l:

        mid = l + (r - l)/2

        # If element is present at the middle itself

        if arr[mid] == x:

            return mid

        # If element is smaller than mid, then it

        # can only be present in left subarray

        elif arr[mid] > x:

            return binarySearch(arr, l, mid-1, x)

        # Else the element can only be present

        # in right subarray

        else:

            return binarySearch(arr, mid + 1, r, x)

    else:

        # Element is not present in the array

        return -1

# Test array

arr = [ 2, 3, 4, 10, 40 ]

x = 10

# Function call

result = binarySearch(arr, 0, len(arr)-1, x)

if result != -1:

    print "Element is present at index % d" % result

else:

    print "Element is not present in array"

# Q6. Provide a definition of Procedural Programming and then a description of two major procedural programming languages. Give a code example of each using binary search algorithm.

Procedural programming uses a linear approach to programming, which is based upon calling procedure. Routines, subroutines and functions are executed in a series of step by step format until the program ends.

### C

C was originally designed to simplify and overcome problems associated with older programming languages such as BASIC and FORTRAN in 1972 by Dennis M. Ritchie. C is an imperative language designed to compile in a simple top down manner and is the father of many modern languages such as C++, C#, Java, Python and many more.

### C Binary Search Algorithm (GeeksforGeeks)

// C program to implement recursive Binary Search

#include <stdio.h>

// A recursive binary search function. It returns

// location of x in given array arr[l..r] is present,

// otherwise -1

int binarySearch(int arr[], int l, int r, int x)

{

    if (r >= l) {

        int mid = l + (r - l) / 2;

        // If the element is present at the middle

        // itself

        if (arr[mid] == x)

            return mid;

        // If element is smaller than mid, then

        // it can only be present in left subarray

        if (arr[mid] > x)

            return binarySearch(arr, l, mid - 1, x);

        // Else the element can only be present

        // in right subarray

        return binarySearch(arr, mid + 1, r, x);

    }

    // We reach here when element is not

    // present in array

    return -1;

}

int main(void)

{

    int arr[] = { 2, 3, 4, 10, 40 };

    int n = sizeof(arr) / sizeof(arr[0]);

    int x = 10;

    int result = binarySearch(arr, 0, n - 1, x);

    (result == -1) ? printf("Element is not present in array")

                   : printf("Element is present at index %d",

                            result);

    return 0;

}

### Pascal

Pascal was developed by Niklaus Wirth in 1970. It was originally developed to teach structured programming and offered advanced programming techniques such as conditional and loop control structures without GOTO statements, and allowed user-defined data types to organise complex information. This made Pascal much more powerful than previous languages such as ALGOL, FORTRAN and COBOL and led to it becoming the most widely used language for programming in the 70’s and 80’s.

### Pascal Binary Search Algorithm (WisdomJobs)

PROGRAM TestBinarySearch;

(\* Test binary search procedure \*)

USES SortEnvLib;

TYPE Range = 0..MaxData;

PROCEDURE BubbleSort(VAR Data: DataArrayType);

(\* Sort array Data in decreasing order \*)

VAR Posn, Temp: INTEGER;

Done: BOOLEAN;

BEGIN

Done := FALSE;

WHILE NOT Done DO BEGIN

Done := TRUE;

FOR Posn := 1 TO MaxData - 1 DO

IF Data[Posn] < Data[Posn + 1] THEN BEGIN

Temp := Data[Posn];

Data[Posn] := Data[Posn+1];

Data[Posn + 1] := Temp;

Done := FALSE;

END;

END;

END; { BubbleSort }

PROCEDURE BinSearch(A: DataArrayType; Low, High: INTEGER;

Key: INTEGER; VAR Mid: INTEGER);

(\* Search decreasingly ordered array A for Key \*)

BEGIN

Mid := (Low + High) DIV 2;

WHILE Low < High DO BEGIN

IF A[Mid] < Key THEN { first half }

High := Mid - 1

ELSE

IF A[Mid] > Key THEN { second half }

Low := Mid + 1

ELSE BEGIN { found }

Low := Mid;

High := Mid;

END;

Mid := (Low + High) DIV 2;

END;

IF A[Mid] <> Key THEN { item not found }

Mid := 0;

END; { BinSearch }

VAR Data: DataArrayType;

SearchKey, Index: INTEGER;

BEGIN

RandomData(Data);

BubbleSort(Data);

WriteLn('Sorted data');

ListData(Data);

WriteLn;

Write('Enter the search key: ');

Read(searchKey);

BinSearch(Data, 1, MaxData, SearchKey, Index);

If Index = 0 THEN

Write('It does not occur ')

ELSE

Write('It is at position ', Index:3);

END. { TestBinarySearch }

# Q7. Provide a definition and description of real-time programming techniques.

Real time programming creates real time systems, systems that are subjected to real time and run routines and functions based upon real world time deadlines. Real time programming can be broken into two main types, hard real time systems and soft real time systems. Hard real time systems are designed to never miss a deadline, as doing so will have catastrophic effects, programs such as flight control systems are hard real time systems. Soft real time systems will occasionally miss deadlines based upon the priority of inputs, procedures or routines and tardiness may gradually increase to accommodate queues of these functions. A telephony switch system is a type of soft real time system.

# Q8. Provide a description of the software application measurement and estimation methodology outlined by the Consortium for IT Software Quality (CISQ)

There are 3 main areas that the CISQ focuses on when it comes to measurement and estimation methodologies for software development. Each of these categories bears a heavy focus on automation and DevOps workflows.

### Software Sizing

Software Sizing is calculated using two separate standards to identify the functional size of the software, and the changes in the functional and non-functional code during runtime, These are named Automatic Function Points, and Automated Enhancement Points respectively.

The Automatic Function Point Standard (AFP) is generally used to automate the functional sizing of a software product, however it can be used for other functions such as:

* Software quality and productivity analyses
* Cost and resource estimation
* Estimation method calibration
* Data normalisation
* Application package functionality sizing

Automatic Enhancement Points build upon the existing AFP to standards to allow the measurement and sizing of both the functional and non-functional sizing of software products. This additional functionality allows the user to:

* Perform quality and productivity analysis
* Determine the ROI by measuring the functionality sizing against the software requirements
* Calibrate estimation methods
* Normalise software data benchmarks
* Determine the size of an application package by sizing all the code included within the package

### Code Quality

Code quality is measured by 4 separate standards that combine to give an indicator of the robustness of the code. They are

* Security measures the weaknesses and exploitable flaws within the code. The standard contains a total of 36 parent weaknesses and 38 contributing weaknesses that have been derived from various sources such as the [CWE/SANS Institute Top 25 Most Dangerous Software Errors](http://cwe.mitre.org/top25/#Listing).
* Reliability measures the risk of software failure and the stability of the application when performing tasks under unexpected conditions. This includes issues with fault tolerance, recoverability and data integrity of an application
* Performance Efficiency standards measure the resource usage, response time and data storage performance of a software application. The standard is comprised of 15 parent weaknesses and 3 contributing weaknesses from the MITRE CWE (Common Weaknesses Enumeration).
* Maintainability assesses efficiency and efficacy with which the software can be patched, modified or updated with relation to concepts such as modularity, changeability and reusability. The standard collects 29 weaknesses effecting the maintainability of a software product.

### Technical Debt

Technical debt is measured with the Automated Technical Debt standard produced by CISQ. This standard is used to estimate the projected cost of all the accumulative weaknesses and vulnerabilities within a software product. The debt is calculated with measurements on the principal sum (the total costs to fix all structural weaknesses and insecurities within the software) and the interest (weaknesses and vulnerabilities which were not accounted for or able to be measured during production and become apparent on deployment). This dual system analysis allows for accurate costs to be determined in the planning stages of projects in functional business metrics, which helps to plan for future corrective costs when unexpected issues arise within the code.

## Q9. Provide a description of software metrics used for software development.

A software metric is any useful measurable data produced or exhibited by features or characteristics of software applications. This data is important when it comes to understanding and measuring software performance and productivity, understanding process patterns and hardware integration. Software metrics are defined into three categories; Product Metrics, Process Metrics and Project Metrics.

### Product Metrics

Product metrics are used to identify and measure the key features or identifying factors within a software product. These include but are not limited to size, complexity, performance, and quality.

### Process Metrics

Process metrics usually consists of data that is collected at runtime and is often used to improve the functionality of the software. Process metric data includes response time data, cycle time, error rate, takt time and throughput.

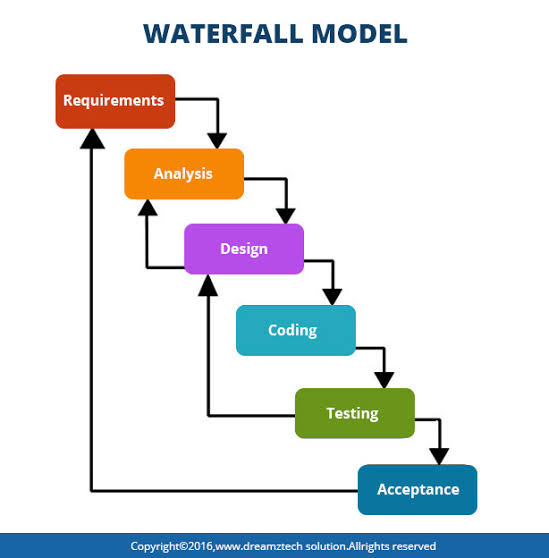
### Project Metrics

Project metrics are the collection of information pertaining to the project itself, rather than the subject of the project. This data can help to analyse business critical information regarding the project and aid with planning and executing a software development project. This data can include planned value data, actual costings, earned values, cost variance, return on investment (ROI), and staffing data.

## Q10. Provide a definition and description of three contemporary industry software development methodologies. Include a diagram of each SDLC.

### Waterfall Model

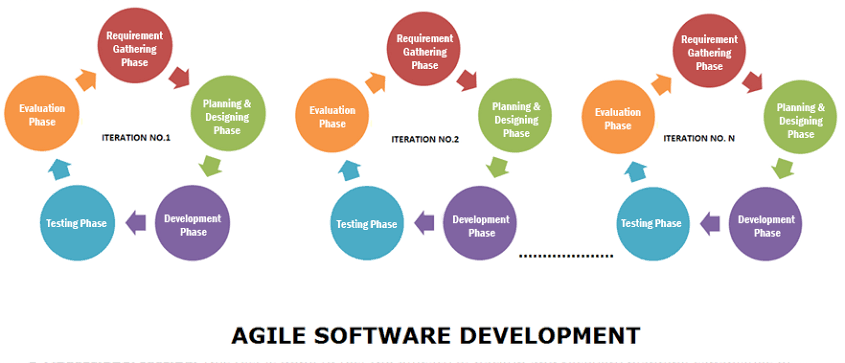
The waterfall model is the most basic and traditional of all SDLC’s. It uses a step by step procedure to work through each phase of the development process sequentially until all phases are complete and the project is ready for deployment. The waterfall model generally comprises of five or six steps; Requirements and Documentation, Analysis, Design, Build, Testing and finally Deployment.



### Agile Model

Agile modelling is a mode of software development that focuses on iterative cycles to build a software product. Agile is superior to Waterfall projects in many ways, most notably that it reduces risk by allowing much more client feedback and testing of the product and is therefore much more suitable for a range of project sizes. This can however lead to excessive input by the customer, or gold plating on part of the developers and can easily be led astray from the original requirements of the project, increasing costs and completion timeframes for the developer. To combat this, small meetings at the beginning of each sprint (the iterative cycle) called scrums are organised to keep staff on task and ensure that the project stays on task.

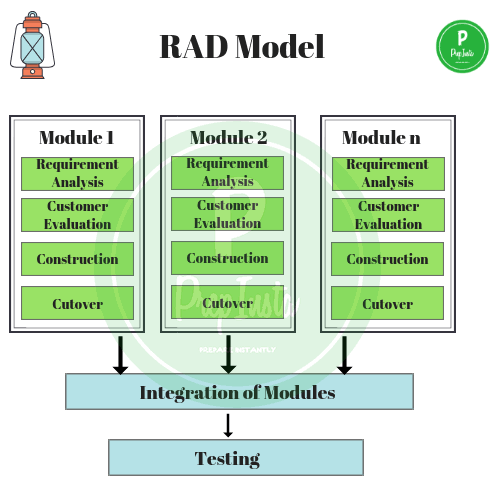
Each iteration or sprint involves cross functional teams working on various aspects of the project simultaneously in order to complete a section or piece of the project, before moving onto the next cycle. Typically, each sprint will contain the cornerstones of the waterfall model (planning, analysis, design, build, testing and deployment) at a much smaller scale indicative of the section or part of the project that is currently being worked on.



### RAD Model

Rapid Application Development is an agile based methodology that focuses on producing workable software prototypes with each sprint which can be integrated into the main build of the project at completion of each sprint or iterative cycle. RAD development is mainly used for applications that can be built from modular pieces of software, which are then packaged into a complete product at the end of the life cycle. It is very customer focused and alleviates a lot of the risk associated with traditional project development methodologies by providing the customer with working prototypes of the software.

Each sprint typically breaks the project down into a portion that can be modularised, if the project elements require a high amount of interdependency, RAD will probably not be suitable for the development of the product. Beyond this, RAD development follows a very similar iterative process to AGILE development, with greater emphasis on the delivery of working software modules at the end of each cycle.



## Q11. Provide a definition of software development and configuration management processes.

Software configuration management (SCM) is a set of processes, procedures, tools and policies employed during software development to organise the development process and maintain the current software baseline. This allows new features and functionality to be added to software without damaging or fragmenting the source code into unreadability or creating avoidable bugs or errors. The goal of SCM is to prevent these errors and fixing them quickly and efficiently when they occur.

SCM uses management patterns to accomplish goals and create organisational charts for development tasks. Each pattern is representative of a certain step in development and are divided into 3 categories.

### Core Patterns

Core patterns are used to define the baseline of the software. This gives a starting point for the development team to work from and measure progress against. The primary pattern in the core structure is the main line, which is the main code that will be used when the product is ready for deployment. To ensure that the main line isn’t destroyed or effected negatively by development changes, a second draft line is created called the active development line. This code is a collection of all the working code modifications from the development team and mitigates risk against the main line source code.

### Workspace Patterns

Workspace patterns consist of the bulk of the development work. These patterns are drip fed through and integrated into the active development line where they are tested and eventually released as a new main line build. The primary workspace pattern is the private workspace pattern. This is where the main line is replicated, and changes are made to the source code without disrupting or affecting other developers working on the same main line source code. Other patterns are also fed into the private workspace pattern which assist with quality control during the development process. These are:

* Private builds
* Integration builds
* Smoke tests
* Unit tests
* Integration tests

### Codeline Patterns

Codeline patterns, also known as branches are utilised when development could alter the mainline and cause errors, or effect development deadlines. These should generally be the least used of all SCM patterns and should adhere strictly to codeline policies and practices. These patterns include but are not limited to

* Earmarking current software versions
* Working on large significant changes that could destabilise the current build
* Leapfrogging and unfinished current release to work on a new release

Codeline policies should be heavily relied upon when using codeline patterns. These policies help ensure that testing and integration processes are maintained before code is released to build iterations and mitigate risks against the project. It is advisable to automate these requirements as much as possible.

## Q12. Provide a description of quality assurance practices and standards as they relate to Rapid Application Development that could be used by CITE Managed Services.

Software quality assurance (SQA) is an ongoing process that is undertaken during the development of a software application to ensure that it meets certain criteria, both specified and unspecified is planning documentation, to ensure that the product works effectively under a variety of conditions. SQA is managed with a quality assurance plan that is created and conducted from the initial stages of planning during the project life cycle and provides an SQA plan document for the development team to utilise and work from until the project has been completed. The document should be measured against two key concepts for quality assurance; validation and verification. Validation based quality assurance practices are applied to ensure that all predefined requirements are met throughout all stages of the project, and verification practices measure the efficacy, reliability and stability of the application by assessing data metrics to confirm the programs credibility and accuracy. Using both concepts are key when developing an SQA plan to ensure that robust quality control measures are applied.

An SQA plan should contain the following measures:

* Checkpoints
* Software Engineering Techniques
* Formal Technical Reviews
* Multi-Testing Strategy
* Process Adherence Enforcement
* Change Control
* Change Impact Measurements
* SQA Audits

The ISO 9000 standards are produced and regulated by the International Organisation for Standardisation for the express purpose of SQM. These standards can be used to aid organisations in delivering quality software that adhere to SQA guidelines and processes.

## Q13. Provide a description of Rapid Application Development and outline the role of the client, and their involvement in each phase of RAD.

Rapid application development follows a procedural incremental based development cycle, spawned from AGILE methodology. The project is broken down into smaller modular pieces, and development begins in “sprints”, short periods of time in which development of the piece of software is completed and tested, and returned to the client for sign off as a prototype which demonstrates the functionality of the product. When the client signs off on the new iteration, it is integrated into the core build and a new RAD phase begins.

## Q14. Describe the client business domain, and then outline the impact it will have on RAD; in particular the effect on the overall cost and quality of the final application.

The client business domain is a set of classes that represent the major operational functions within a business. These classes are organised to contain the information and behaviour that are required for the business to operate and achieve the goals of the business. In a RAD environment, the business domain will be divided into separate sets of objects, and each one will follow the development life cycle phases until they are completed. Broader business domains will require more iterations, and generally will drive the cost of the project upward.

## Q15. Create a comparison table of contemporary RAD tools, add the table headings like: targeted platform, RAD Type, cost, rating and major benefits. Your table should have more than 5 RAD tools.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Targeted Platform | RAD Type | Cost | Rating | Major Benefits |
| Google App Maker | Small and Large Scale – Cloud Hosted | $10 Monthly | 100% | Easy to use drag and drop functions  Built in tutorials  Google Analytics Integration |
| GitHub | Small, medium, large and personal scale – Cloud hosted | $7 Monthly | 98% | Seamless Code Review  Pull and Merge Requests  Personalised Project Management Tools |
| Microsoft PowerApps | Small, medium and large scale – Cloud and on premises hosting | $7 Monthly or Quote Based | 97% | App Creation and Running  App Sharing  Cloud Based Services Integration |
| Salesforce Lightning | Small, medium and large scale – Cloud hosted | Quote Based | 100% | Interactive Tabs  Activity timeline  Homepage components  Path  The Assistant |
| BitBucket | Small, medium and personal scale – Cloud and on premises hosting | Free or additional features for $2 monthly payment | 96% | Code Review  Branch Permissions  Pipelines |

# References

Carl Cahill, L. M. (2018, February 20). *12 top prototyping tools*. Retrieved from Creative Bloq: https://www.creativebloq.com/web-design/top-10-prototyping-tools-2016-21619216

Chand, M. (2019, April 27). *What Is the Most Popular Operating System?* Retrieved from C# Corner: https://www.c-sharpcorner.com/article/what-is-the-most-popular-operating-system/

Cisco Start. (2019). *How to Set Up a network for Small Businesses*. Retrieved from Cisco: https://www.cisco.com/c/en\_au/solutions/small-business/resource-center/networking/primer-building-small-office-network.html

GeeksforGeeks. (2019). *Binary Search*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/binary-search/

Grigg, C. (2019, July 26). *Types of Programs Found in Office Productivity Suites*. Retrieved from lifewire: https://www.lifewire.com/programs-found-in-office-productivity-suites-2511774

Hemmendinger, D. (2018, December 28). *Pascal Computer Language*. Retrieved from Britannica: https://www.britannica.com/technology/Pascal-computer-language

Home Stratosphere. (2019). *46 Types of Office Equipment (Checklist) for a Fully Functional Office*. Retrieved from Home Stratosphere: https://www.homestratosphere.com/types-of-office-equipment/

Khanna, M. (2019). *Different Types of Office Layouts*. Retrieved from Commercial Property 2 Sell: https://www.commercialproperty2sell.com.au/blog/2018/03/different-types-of-office-layouts.php

OpenSource. (2019). *What is Linux?* Retrieved from opensource.com: https://opensource.com/resources/linux

Rouse, M. (2019, March). *device driver*. Retrieved from SearchEnterpriseDesktop: https://searchenterprisedesktop.techtarget.com/definition/device-driver

Rowley, D. (2019, February 19). *Data storage: Which of the 3 types is best for your business?* Retrieved from Big Data - Made Simple: https://bigdata-madesimple.com/data-storage-which-of-the-3-types-is-best-for-your-business/

South Metro Tafe. (2019). *Business Domain*. Retrieved from Blackboard - South Metro Tafe: https://blackboard.southmetrotafe.wa.edu.au/bbcswebdav/pid-1728971-dt-content-rid-15806270\_1/courses/16\_C\_EIO\_BUSAPPCLU\_1/Business\_Domain.pdf

Tutorials Point. (2019). *SDLC - Agile Model*. Retrieved from LEARN SDLC: https://www.tutorialspoint.com/sdlc/sdlc\_agile\_model.htm

Tutorials Point. (2019). *SDLC - RAD Model*. Retrieved from LEARN SDLC: https://www.tutorialspoint.com/sdlc/sdlc\_rad\_model.htm

Tutorials Point. (2019). *SDLC - Waterfall Model*. Retrieved from LEARN SDLC: https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm

W3Schools. (2019). *History of C Programming Language*. Retrieved from W3Schools: https://www.w3schools.in/c-tutorial/history-of-c/

WisdomJobs. (2018). *BINARY SEARCH - PASCAL PROGRAMMING*. Retrieved from wisdomjobs: https://www.wisdomjobs.com/e-university/pascal-programming-tutorial-168/binary-search-7074.html