ASP.NET Web Application

Group Members

**Ben – 5101965116**

**Michael - 3100553617**

**Corey - 0100601817**

**Waleed - 6100758617**

**Andrei Rico – 3106107616**

Contents

[Out-line of Current Software Development Methodologies - Corey: 3](#_Toc498585346)

[Description of Projects Development Cycle - Andrei: 3](#_Toc498585347)

[Explanation of Object-Oriented Programing - Waleed: 6](#_Toc498585348)

[Description of open-source development tools you could have used in the project - Ben: 6](#_Toc498585349)

[Description of software-testing used in development of the project - Micheal: 6](#_Toc498585350)

# Out-line of Current Software Development Methodologies - Corey:

Some currently used software development methodologies which are commonly used include:

Rapid Application Development (RAD)

This methodology aims to produce decent quality production within a limited time. RAD may not be the best suited development method for more complex tasks, it tends to suit in-house and limited-distribution software’s.

Waterfall (Traditional)

Water, or traditional development is a methodology which works with strict deadlines and phases of the development cycle. The program will be split out into different parts and completed in order, once a phase has been completed the developers will not return to that part. This development cycle is commonly used but due to its linear model certain problems can occur, such as returning to the program or certain parts to fix making it undesired for certain task.

Lean Development (LD)

Lean Development is a very business focused development method, the cycle works with 12 core principles with the first being ‘Satisfying the customer is the highest priority.’ This development is change-tolerant and focus on completing the task and not constructing. The Lean development works to reduce time committed to complete a task, using the principle of minimalism is essential.

# Description of Projects Development Cycle - Andrei:

* **A detailed description of the project’s software development life cycle (SDLC)**

**SDLC IN DETAIL**

* Project Planning
  + Prepare
  + Review
  + Rework
  + Baseline
  + Revise [if necessary] >> Review >> Rework >> Baseline
* Requirements Development [Business Requirements and Software/Product Requirements]
  + Develop
  + Review
  + Rework
  + Baseline
  + Revise [if necessary] >> Review >> Rework >> Baseline
* Estimation [Size / Effort / Cost]
  + <same as the activities/tasks mentioned for Project Planning>
* Scheduling
  + <same as the activities/tasks mentioned for Project Planning>
* Designing [ High Level Design and Detail Design]
  + <same as the activities/tasks mentioned for Requirements Development>
* Coding
  + Code
  + Review
  + Rework
  + Commit
  + Recode [if necessary] >> Review >> Rework >> Commit
* Test Builds Preparation/Deployment
  + Build/Deployment Plan
    - Prepare
    - Review
    - Rework
    - Baseline
    - Revise [if necessary] >> Review >> Rework >> Baseline
  + Build/Deploy
* Unit Testing
  + Test Plan
    - Prepare
    - Review
    - Rework
    - Baseline
    - Revise [if necessary] >> Review >> Rework >> Baseline
  + Test Cases/Scripts
    - Prepare
    - Review
    - Rework
    - Baseline
    - Execute
    - Revise [if necessary] >> Review >> Rework >> Baseline >> Execute
* Integration Testing
  + <same as the activities/tasks mentioned for unit testing>
* User Documentation
  + Prepare
  + Review
  + Rework
  + Baseline
  + Revise [if necessary] >> Review >> Rework >> Baseline
* System Testing
  + <same as the activities/tasks mentioned for Unit Testing>
* Acceptance Testing[ Internal Acceptance Test and External Acceptance Test]
  + <same as the activities/tasks mentioned for Unit Testing>
* Production Build/Deployment
  + <same as the activities/tasks mentioned for Test Build/Deployment>
* Release
  + Prepare
  + Review
  + Rework
  + Release
* Maintenance
  + Recode [Enhance software / Fix bugs]
  + Retest
  + Redeploy
  + Rerelease

# Explanation of Object-Oriented Programing - Waleed:

**Answer:** It is a programming model where programs are organized around objects and data rather than action and logic. Object means a real word entity such as pen, chair, table etc. It simplifies the software development and maintenance by providing some concepts: Object, Class, Inheritance, Polymorphism, Abstraction, and Encapsulation. OOPs provides many applications such as Real Time Systems, Simulation & Modelling, and Decision Support System.

# Description of open-source development tools you could have used in the project - Ben:

Various development tools were used during the creation of the software, a list of these tools can be found directly below:

* Microsoft Visual Studio 2015
* Microsoft Visual Studio 2017
* ASP.NET
* Microsoft SQL Server 2016

# Description of software-testing used in development of the project:

To test the software and make sure it was working as it was intended – we would simply just run the program and go and debug the code to make sure it was working the way we intended it too.

If something wasn’t working as it was intended we would fix the issues and try again.