

Four software development methodologies



**ASSIGN
BUSTER**

Student Name Student ID Student Signature Assessor Name Assessor's Signature Final Result / Grade Assessed By / Date Case Study Research (worth 20% of final grade) 100 Marks SOF535: Application Design and Development Version 1. 1, Level 5, Credits 20 New Zealand Diploma in Information Technology Technical Support Level 5 Task 1: Requirement Gathering and Analysis (Total 40 Marks) comparison between four software development methodologies.

Four Software Development Methodologies Waterfall Development Methodology The Waterfall approach to systems analysis and design was the first established modern approach to building a system. This method was originally defined by Winston W. Royce in 1970, (" The Waterfall Development Methodology", 2006).

It quickly gained support from managers because everything flows logically from the beginning of a project through the end, (Jonasson, 2008). Sources differ when it comes to the specific steps in the Waterfall process.

Rapid Application Development Methodology

Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product. In the RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery.

Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process. RAD projects follow iterative and incremental model and have small teams comprising of developers, domain

experts, customer representatives and other IT resources working progressively on their component or prototype. The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable.

Agile Development Methodology

There are many different forms of the Agile development method including: Scrum, Crystal, Extreme Programming (XP), and Feature-Driven Development (FDD). Agile methods attempt to minimize risk (such as bugs, cost overruns, and changing requirements) by developing the software in iterations that are mini-increments of the new functionality.

The benefit of multiple iterations is that it improves efficiency by finding and fixing defects and expectation mismatches early on. Agile methods rely on real-time communication, which fails to provide new users with documentation to get up to speed. They require a huge time commitment from the users, and is labor-intensive on the developers due to the need to fully complete each feature within each iteration for user approval. The benefits to the software are realized early on due to the incremental addition of the iterations.

The Agile methods are similar to RAD, and can be inefficient in large organizations. Programmers, management and organizations accustomed to the Waterfall Method may have difficulty adjusting to Agile, and often a hybrid approach works well for them.

DevOps Deployment Methodology

DevOps deployment is centered around organizational change that enhances the collaboration between the departments responsible for different segments of the development life cycle, such as development, quality assurance and operations. It is focused on improving the time to market, lowering the failure rate of new releases, shortening the lead time between fixes, and prioritizing minimal disruption as well as maximum reliability.

To achieve this, DevOps aims to automate its continuous deployment to ensure everything happens smoothly and reliably. Companies that use DevOps have benefitted by significantly reducing the time to market, improving customer satisfaction, improving product quality and improving productivity and efficiency of its employees.

A few drawbacks of DevOps include appropriate methodology that is suitable for this project. It totally depends on the nature of the project. Many people might argue that Agile is the best methodology for software development and it does have significant advantages in most cases; however, saying "Agile is better than Waterfall" is like saying "A car is better than a boat".

They both have advantages and disadvantages depending on the environment. Agile/Scrum works best with projects that have high levels of uncertainty and require an adaptive approach. A plan-driven approach (what many people loosely call "Waterfall") works best in situations where the requirements are fairly certain and there is some need for predictability of project costs and schedules. And, of course, that is not meant to imply that there is a binary and mutually-exclusive choice between two extremes.

There are many ways to blend an adaptive (Agile) approach with a plan-driven approach in the right proportions to fit the situation. There is no "silver bullet" methodology that fits all possible projects and the right solution is to fit the methodology to the nature of the problem rather than force-fitting a project to some predefined methodology. <https://blog.blackducksoftware.com/top-4-software-development-methodologies>

Two types of requirements

1. Functional Requirements

Functional requirements define the fundamental actions that system must perform. The functional requirements for the system are divided into three main categories, Reservation/Booking, Food, and Management. For further details, refer to the use cases. EXAMPLE 1. Reservation/Booking
The system shall record reservations. The system shall record the customer's first name. The system shall record the customer's last name.

The system shall record the number of occupants. The system shall record the room number. The system shall display the default room rate. The system shall allow the default room rate to be changed. The system shall require a comment to be entered, describing the reason for changing the default room rate. The system shall record the customer's phone number. The system shall display whether or not the room is guaranteed.

The system shall generate a unique confirmation number for each reservation. The system shall automatically cancel non-guaranteed reservations if the customer has not provided their credit card number by 6:00 pm on the check-in date. EXAMPLE 2 Food
The system shall track all meals purchased in the hotel (restaurant and room service).

The system shall record payment and payment type for meals. The system shall bill the current room if payment is not made at time of service. The system shall accept reservations for the restaurant and room service.

Management

The system shall display the hotel occupancy for a specified period of time (days; including past, present, and future dates). The system shall display projected occupancy for a period of time (days). The system shall display room revenue for a specified period of time (days). The system shall display food revenue for a specified period of time (days).

The system shall display an exception report, showing where default room and food prices have been overridden. The system shall allow for the addition of information, regarding rooms, rates, menu items, prices, and user profiles. The system shall allow for the deletion of information, regarding rooms, rates, menu items, prices, and user profiles. The system shall allow for the modification of information, regarding rooms, rates, menu items, prices, and user profiles. The system shall allow managers to assign user passwords.