|  |
| --- |
| Creative Instinct Building Supply |
| Deployment Phase |
| Capstone Project |

|  |
| --- |
| Miguel Strother  12-20-2022 |

**Table of Contents**

2 Introduction

3 UAT

3 Biblical principles

4 Screenshots

6 Deployment Plan

11 References

According to Mäntylä & Vanhanen (2011), “software deployment is a set of critical activities for all software vendors, from an order for a new software requirement to the necessary measures for a new version available to the customer. This deployment is composed of activities that are essential to make a product available, such as: installation of dependencies, configuration files and installation of the application itself.” The Deployment Phase includes the necessary work to deliver the final deliverables of the system to the sponsor. It provides production installation and customer acceptance which requires all software execution testing to be correct.

In the Deployment Phase the transition is made from the legacy system and the new system is rolled out. According to Sommerville (2016), “during the deployment, errors in system configuration can generate new vulnerabilities, that can lead to system operations errors. When making changes to the system, some considerations made during the original purchase can be forgotten and again, vulnerabilities can be introduced into the system.” This is why identifying the risk factors is extremely important during the deployment phase. The number one cause of risk factors is minimal concern when testing. “Table 8 provides a list of the identified risk factors and the responses of the respective employees, based on the probability of their occurrence. Table 8 shows that the risk factors mentioned by most respondents are related to: (1) Problems during implantation with a 90% index; (2) Lack of Resources and Test Failures with an index of 80%, and (3) Difficulty in using the System and User Resistance to Changes with an index of 70%,” (Jistem USP, 2022).

Table

Description automatically generated with low confidence

External changes may also play a role as a risk factor. “The main reason for software delays and cancellations is due to the large number of errors, the elimination of which can absorb more than 60% of the effort in large software projects. The North American average for removing software defects is 92.5%, referring to the 2017 period. In the implementation phase, the removal of defects in the beta and acceptance tests corresponds to approximately 20% (Jones, 2017).”

UAT is important because it allows the end users to test the new system and give feedback. It also allows the developers to fix any bugs or issues in the system such as patches that may be needed or code changes. We used user stories that relate to the business requirements to develop the acceptance criteria. We used Beta testing to give end users an opportunity to evaluate the system and give feedback to the team. The participation of end users has a great impact on the implementation of the software, which can lead to failures or success in the process. “It is common to have resistance to changes during their execution, especially if they are imposed externally, which can have a negative influence on the final result of the project,” (Sun-Jen & Wen-Ming, 2008). The UAT results found that checking the product selection feature, most end users did not like the concept of having to input the buyers name with the product. This made it more tedious for contractors and clients paying while purchasing multiple products. Also, the sponsor/customer decided that sales were down due to Covid-19, and since they were being over charged for certain products, they would stick to just selling roofing shingles. So, we decided to revamp the UI to make it a little easier since some end users were having issues adapting to the change. Our product weighs well against the CIS Benchmarks for IT technologies and systems. Our Microsoft Windows Server configurations were in line with IS best practice configurations. Our network settings from our Amazon Web Services cloud provider were in compliance with regulations and IT governance as well.

We will roll out our new system using the pilot operation changeover. After the system proves successful the direct cutoff method will be implemented.

It is important to ensure that God is always first. Just like this project, everything we do, we should do in God’s name. This project is in accordance with his purpose for His people. When we think of deployment we think of military going to other countries to serve, but deployment can mean developed followers of Christ going out to fulfill God’s purpose. Jesus’ message in Matthew 28: 19-20, before he ascended into heaven was to go out baptize and make disciples of all nations, and basically teach the world. First John 4:13-14 says, “By this we know that we abide in him and he in us, because he has given us of his Spirit. And we have seen and testify that the Father has sent his Son to be the Savior of the world.” We’ve seen it and now we testify that Jesus is the Savior of the world. That’s why Paul can write in 2 Corinthians 5:20, “Therefore, we are ambassadors for Christ, God making his appeal through us. We implore you on behalf of Christ, be reconciled to God.” So all of God’s people are called to go out with the message of the gospel and proclaim Jesus Christ, because the Kingdom grows through disciples who are deployed. It was recorded in the new testament that the Apostle Paul started at least 14 churches, all while facing tremendous hardships. Another example of deployment to deliver the Word is the story of Hudson Taylor who was a missionary in China some hundred years ago. He contributed 51 years of missionary service and established 20 mission stations. He trained 700 Chinese workers who helped raise four million dollars, and developed a Chinese church congregation of 125,000 Christians. We are deployed to make a difference and remain humble at the same time. Just as in the deployment phase of a project we must “test and show ourselves approved,” by testing our system, remaining receptive and teachable if issues arise, and put forth the work to deliver a quality product.

Graphical user interface, text, application

Description automatically generated

Deployment Plan

Creative Instinct Building Supply

Capstone Project

Miguel Strother

Liberty University

Table of Contents

1. Introduction 7

2. Deployment Planning 7

2.1 Responsibilities 7

2.2 Schedule 7

3. Resources 7

3.1 Facilities 7

3.2 Hardware 7

3.3 The Deployment Unit 8

3.3.1 Support Personnel 8

4. Training 8

5. Deployment Strategies 8

5.1 Risks and Concerns 8

5.2 Contingencies 8

5.3 Compatibility, Conversion, and Migration Strategies 8

6. Deployment Management 8

6.1 Problem Tracking Process 8

6.2 Problem Reporting 8

Creative Instinct Building Supply Deployment Plan

Introduction

This document contains some of the high-level tasks and considerations that will be addressed during the deployment of the new system for Creative Instinct Building Supply.

# Deployment Planning

## Responsibilities

It is the responsibility of the product design group to ensure that all prototypes meet the customers specifications and requirements. The development team are proficient software engineers who will develop and pre-QA test all code generated. They will be lead by our team leaders who are knowledgeable in software infrastructure tools and SDLC. UAT testing will be the last phase of the software testing process. The end users play a vital role because they will interact with the system to see if any features have been overlooked, to see if the UI is smooth and easy to navigate, and see if there are any bugs that may need to be debugged or patches that may need to be applied.

## Schedule

This section is a schedule of milestones to conduct the deployment activities.

* Planning the deployment
* Developing support material
* Managing acceptance tests
* Acceptance testing (integration, unit, and systems testing)
* Revisit Scope, schedule, and budget
  + Update schedule and budget
* Producing the deployment system
* Produce performance reports
* Monitor stakeholders satisfaction
* Manage and mitigate risks
* Deploy final product

# Resources

## Facilities

We have outsourced our testing to Zenergy Technologies. Their facility is equipped with encryption, firewalls, and IPS to ensure quality assurance during software testing on functionality and user experience. Their cost effective team will come test during different iterations throughout the entire SDLC.

## Hardware

A 64-bit environment is recommended with at least 250GB to check the code and 150GB to build.

## The Deployment Unit

### Support Personnel

Our technical team will be available via chat, email, or phone call 24/7.

# Training

End users will have the opportunity to train on the new system and UI via YouTube videos, live chat online, and webinar. We will also have a question-and-answer session that will be recorded for later use if necessary.

# Deployment Strategies

## Risks and Concerns

The risk we have received from feedback from our end users is ease of use on UI.

## Contingencies

Any problems encountered during the delivery of the product are reported to the contact identified on the deployment request.

## Compatibility, Conversion, and Migration Strategies

Our product weighs well against the CIS Benchmarks for IT technologies and systems. Our Microsoft Windows Server configurations were in line with IS best practice configurations. Our network settings from our Amazon Web Services cloud provider were in compliance with regulations and IT governance as well.

# Deployment Management

## Problem Tracking Process

Project development teams will use ClearQuest to manage all aspects of Configuration and Change Management for the project.

## Problem Reporting

Issues encountered throughout the deployment phase are reported to the project development team; it is their responsibility to record and track reported issues.

Mäntylä, M.; Vanhanen, J. Software deployment activities and challenges - a case study of four software product companies. In: Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on., 2011. p. 131–140. ISSN1534-5351

Sommerville, I. (2016). Software Engineering, Addison Wesley

Jones, C., (2017). Exceeding 99% in Defect Removal Efficiency (DRE) for Software., Technical Report, Software Estimation Journal; Namcook Analytics; Saint John – AB.

The Bible. (2011). *New International Version.* Retrieved from: <https://www.biblegateway.com/>

Sun-Jen, H.; Wen-Ming, H.; (2008). Exploring the relationship between software.

Journal of Information Systems and Technology Management-Jistem USP. (2022). *RISK FACTORS IN THE SOFTWARE DEPLOYMENT PHASE: A CASE STUDY APPLIED IN TWO BRAZILIAN GOVERNMENT COMPANIES Vol. 19.* Retrieved from: <https://www.scielo.br/j/jistm/a/zMkdKhfH8jJgMF6hbQbtRHb/?format=pdf&lang=en>