Table of Contents

Introduction and Objectives 3
Project Purpose 3
Key Objectives 3
Expected Benefits 3
Current CodeIgniter Environment Analysis 4
CodeIgniter Version and Configuration 4
Customizations and Third-Party Modules 4
Current Limitations and Challenges 4
Impact of Limitations 5
Upgrade Scope and Version Targeting 5
Impact and Compatibility 5
Backward Compatibility 5
Technical Assessment and Compatibility Analysis 6
PHP Requirements and Extensions 6
Module and Library Compatibility 6
Handling Deprecated Features 6
Compatibility Risk Assessment 7
Upgrade Implementation Plan and Timeline 7
Upgrade Phases 7
Roles and Responsibilities 8
Detailed Timeline and Milestones 8
Dependencies and Approvals 8
Grant Chart 8
Testing and Quality Assurance Strategy 9
Test Case Coverage 9
Performance and Security Validation 9
Testing Tools and Methodologies 10
Risk Assessment and Mitigation Measures 10
Potential Risks 10
Mitigation Strategies 11
Fallback and Rollback Procedures 12
Rollback and Contingency Plan 12
Rollback Triggers 12

- Rollback Procedure 12
- Communication 13

Post-Upgrade Monitoring and Maintenance 13

- Monitoring Processes 13
- Issue Tracking and Escalation 13
- Ongoing Maintenance 14

Summary and Approval 14

- Approval Request 14
- Signatures 14

Introduction and Objectives

This document outlines Docupal Demo, LLC's proposal to update or upgrade the CodeIgniter framework for Acme, Inc's web application. ACME-1 will realize significant improvements to its website through this project. Our primary focus is to deliver enhanced security and improved performance.

Project Purpose

The purpose of this project is to transition ACME-1's existing web application to a more modern and secure version of the CodeIgniter framework. This initiative directly addresses the need for improved website speed, a stronger security posture, and reduced ongoing maintenance costs.

Key Objectives

The successful completion of this CodeIgniter update/upgrade project will achieve the following key objectives for ACME-1:

- **Enhanced Security:** Implement the latest security features available in the updated CodeIgniter framework. This will reduce the website's vulnerability to potential attacks and safeguard sensitive data.
- **Improved Performance:** Optimize the web application's codebase to achieve faster page load times and an overall improved user experience.
- **Reduced Maintenance Overhead:** Streamline the codebase and leverage modern framework features to simplify future maintenance and updates. This will lead to lower long-term operational costs.

Expected Benefits

By achieving these objectives, ACME-1 will realize tangible benefits, including:

- Faster page load times, leading to improved user engagement and satisfaction.
- A reduced risk of security breaches and data compromises.
- Easier code maintenance, allowing for quicker bug fixes and feature implementations.

Current CodeIgniter Environment Analysis

ACME-1 currently operates on CodeIgniter version 3.1.10. The server environment uses PHP 7.2 and MySQL 5.7. This analysis outlines the strengths and weaknesses of the existing setup.

CodeIgniter Version and Configuration

The core framework is CodeIgniter 3.1.10. While stable, this version is several years old. Newer CodeIgniter versions offer improved features, security enhancements, and better PHP compatibility.

Customizations and Third-Party Modules

ACME-1 utilizes a custom authentication module. This module requires careful review during any upgrade to ensure continued functionality. In addition to the custom module, several third-party libraries are integrated. A detailed audit of these libraries is needed to assess compatibility with newer CodeIgniter and PHP versions.

Current Limitations and Challenges

The existing environment presents several challenges:

- **Outdated PHP Version:** PHP 7.2 has reached its end of life. This means it no longer receives security updates. Using an outdated PHP version exposes ACME-1 to potential security vulnerabilities. Upgrading to a supported PHP version is crucial.
- **Slow Database Queries:** Performance bottlenecks exist due to slow database queries. Analyzing and optimizing these queries is essential for improving application responsiveness.
- **Security Practices:** The current setup lacks support for modern security practices. Newer CodeIgniter versions and updated libraries offer improved security features.

Impact of Limitations

These limitations can impact ACME-1 in several ways:

- **Security Risks:** Outdated software increases the risk of security breaches and data compromise.
- **Performance Issues:** Slow database queries lead to poor user experience.
- **Maintenance Challenges:** Maintaining an outdated system is increasingly difficult and costly.
- **Missed Opportunities:** ACME-1 cannot take advantage of new features and improvements in newer CodeIgniter and PHP versions.

Upgrade Scope and Version Targeting

This upgrade targets CodeIgniter version 4. The project will migrate ACME-1's existing application to this latest version. This ensures ACME-1 benefits from the framework's improvements. These include enhanced security features and better performance.

Impact and Compatibility

The upgrade to CodeIgniter 4 involves breaking changes. Therefore, migration scripts are essential to ensure a smooth transition. Docupal Demo, LLC will develop and implement these scripts. This will address the structural and functional differences between the current version and CodeIgniter 4.

Several components require specific updates for compatibility. The custom authentication module needs modification. This ensures it integrates correctly with the new framework version. We will also update third-party libraries. This guarantees they function as expected within the upgraded application.

Backward Compatibility

Complete backward compatibility with the existing system is not feasible due to the core changes in CodeIgniter 4. We will prioritize maintaining data integrity and critical functionalities. We will provide detailed documentation outlining the changes. This will help ACME-1 adapt to the upgraded environment. Thorough testing will also be conducted during and after the upgrade process. This minimizes disruptions and ensures a stable, updated application.

Technical Assessment and Compatibility Analysis

This section outlines our technical assessment for upgrading ACME-1's CodeIgniter framework. It covers the necessary requirements, potential compatibility issues, and how we plan to address them.

PHP Requirements and Extensions

The updated CodeIgniter version requires PHP 7.4 or later. The intl, json, and mbstring extensions must also be enabled on the server. Our initial assessment confirms that the current server meets the minimum PHP version requirement. We will verify that all required extensions are enabled before proceeding with the update. If extensions are missing, we will install and configure them.

Module and Library Compatibility

We have identified potential compatibility issues with third-party libraries used for image processing and PDF generation. To address this, we will conduct thorough testing of these libraries with the new CodeIgniter version. We will also check for updated versions of these libraries that are compatible with CodeIgniter 4. If necessary, we will explore alternative libraries that offer similar functionality and are compatible.

Handling Deprecated Features

CodeIgniter 4 introduces changes that deprecate some features found in earlier versions. We will identify all instances of deprecated features in ACME-1's existing codebase. For each deprecated feature, we will implement the recommended alternative provided by CodeIgniter 4. This ensures that the application remains functional and takes advantage of the improvements offered by the new version.

Compatibility Risk Assessment

The following chart shows the risk assessment regarding the compatibility of different modules:

The chart illustrates potential compatibility risk levels across different application components. "Core" represents the CodeIgniter framework itself, with minimal risk expected. "Libraries" and "Modules" carry moderate risks due to potential third-party compatibility issues. "Database" interactions are also assessed for compatibility with the new framework version. We will closely monitor these areas during the upgrade process.

Upgrade Implementation Plan and Timeline

This section details the plan for upgrading ACME-1's CodeIgniter framework. It outlines the key phases, responsibilities, and timelines. The upgrade will follow a structured approach to minimize disruption and ensure a successful outcome.

Upgrade Phases

The upgrade process includes these phases:

1. **Development:** The existing CodeIgniter application will be analyzed. The code will then be modified to be compatible with the new CodeIgniter version.

2. **Testing:** Rigorous testing will be performed. This includes unit tests, integration tests, and user acceptance testing (UAT).

3. **Deployment:** The upgraded application will be deployed to the production environment.

4. **Post-Deployment Monitoring:** We will actively monitor the application after deployment. This is to identify and resolve any issues quickly.

Roles and Responsibilities

Specific team members will be responsible for each phase:

- **Development:** John Smith
- **Testing:** Jane Doe
- **Deployment:** David Lee

Detailed Timeline and Milestones

Task	Start Date	End Date	Duration	Responsible
Code Analysis	2025-08-19	2025-08-22	4 days	John Smith
Code Modification	2025-08-25	2025-09-05	2 weeks	John Smith
Unit Testing	2025-09-08	2025-09-12	5 days	Jane Doe
Integration Testing	2025-09-15	2025-09-19	5 days	Jane Doe
User Acceptance Testing	2025-09-22	2025-09-26	5 days	Jane Doe
Security Review Approval	2025-09-29	2025-10-03	5 days	IT Security
Database Admin Approval	2025-09-29	2025-10-03	5 days	DBA
Deployment to Production	2025-10-06	2025-10-06	1 day	David Lee
Post-Deployment Monitoring	2025-10-07	2025-10-10	4 days	David Lee

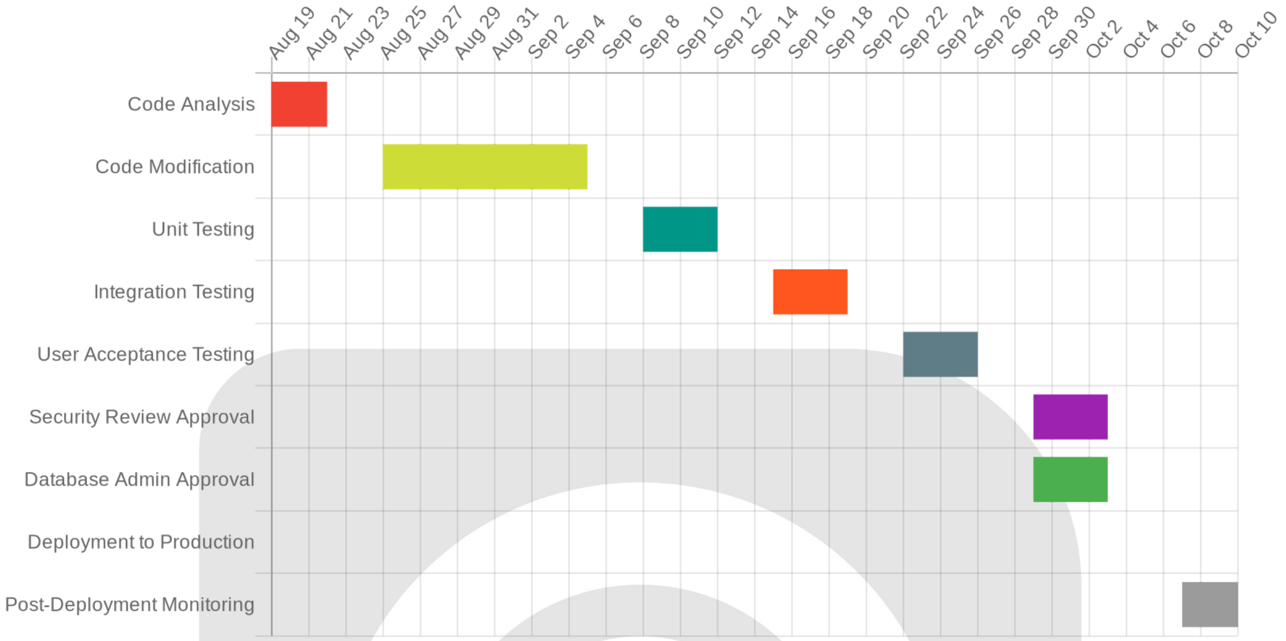
Dependencies and Approvals

The upgrade process depends on the following:

- Approval from the IT Security team.
- Approval from the database administrator.

These approvals are required before deploying to the production environment.

Grant Chart



Testing and Quality Assurance Strategy

We will employ a comprehensive testing strategy to ensure a successful CodeIgniter update/upgrade for ACME-1. This strategy covers various testing levels and methodologies to validate both existing and new functionalities.

Test Case Coverage

Our test cases will cover critical functionalities, including:

- User login processes.
- Data validation procedures.
- Form submission workflows.
- Database interactions.

These test cases will ensure that the core features of ACME-1's application remain stable and functional after the upgrade.

Performance and Security Validation

Post-upgrade, we will validate application performance and security through rigorous testing methods:

- **Performance Validation:** Load testing will simulate user traffic to identify potential bottlenecks and ensure the application's responsiveness under pressure.
- **Security Validation:** Penetration testing will simulate real-world cyberattacks to uncover vulnerabilities and confirm the application's security posture.

Testing Tools and Methodologies

Our testing approach will incorporate a mix of automated and manual testing techniques, utilizing industry-standard tools:

- **Unit Testing:** PHPUnit will be used to perform unit tests on individual code components, ensuring each functions as expected.
- **Integration Testing:** Selenium will automate integration tests to verify the interaction between different parts of the application.
- **User Acceptance Testing (UAT):** ACME-1's designated users will manually test the upgraded application to ensure it meets their requirements and expectations. This includes real-world scenario testing and feedback gathering.

By combining these testing methodologies and tools, we aim to deliver a stable, secure, and high-performing application to ACME-1 post-upgrade.

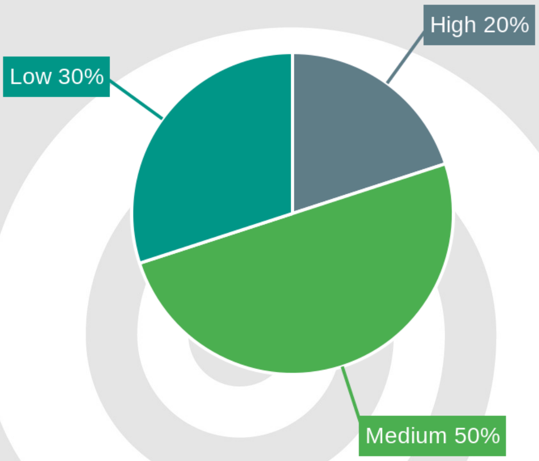
Risk Assessment and Mitigation Measures

This section identifies potential risks associated with the CodeIgniter update/upgrade project for ACME-1 and outlines corresponding mitigation strategies. We have assessed potential technical challenges, data integrity concerns, and deployment-related risks.

Potential Risks

Several factors could impact the success of the upgrade:

- **Compatibility Issues:** Third-party modules may not be fully compatible with the new CodeIgniter version.
- **Data Migration Problems:** Unexpected issues during data migration could lead to data inconsistencies or loss.
- **Downtime During Deployment:** The deployment process may require a period of downtime, impacting ACME-1's operations.



Mitigation Strategies

To minimize the impact of these risks, Docupal Demo, LLC will implement the following measures:

- **Staging Environment Testing:** A complete replica of ACME-1's environment will be created for thorough testing before deployment. This allows us to identify and resolve compatibility issues in a controlled setting.
- **Comprehensive Database Backups:** Full database backups will be performed before any upgrade activity. This ensures that we can restore the system to its previous state if needed.

- **Minimized Deployment Window:** We will optimize the deployment process to reduce the required downtime. This includes careful planning and automation of key tasks.
- **Code Rollback Scripts:** Scripts will be prepared to quickly revert code changes if issues arise after deployment.

Fallback and Rollback Procedures

In the event of a failed upgrade, Docupal Demo, LLC will execute a well-defined rollback plan:

1. **Database Restoration:** Restore the database to the pre-upgrade state using the backups.
2. **Code Rollback:** Execute the code rollback scripts to revert to the previous CodeIgniter version.
3. **Detailed Rollback Checklist:** A detailed checklist will guide the rollback process, ensuring all steps are completed correctly.

Rollback and Contingency Plan

This section outlines the procedures to follow if the CodeIgniter update/upgrade fails or introduces unacceptable issues. The goal is to minimize disruption and restore the system to its previous stable state.

Rollback Triggers

A rollback will be initiated if any of the following conditions are met post-deployment:

- Critical errors that prevent core functionality.
- Significant performance degradation impacting user experience.
- Unresolved security vulnerabilities discovered after the upgrade.

Rollback Procedure

1. **Initiate Rollback:** Upon identifying a rollback trigger, the project lead will immediately initiate the rollback procedure.

2. **Restore Database:** The database will be restored to the latest backup taken before the upgrade. Transaction management will be used to ensure data integrity during the rollback process.
3. **Revert Code:** The application code will be reverted to the previous stable version using our version control system.
4. **Verification:** After the rollback, thorough testing will be conducted to ensure the system is functioning correctly and that data integrity is maintained.

Communication

During the rollback process, clear communication is essential. We will use the following communication channels:

- **Daily Stand-up Meetings:** Daily meetings to discuss progress and any issues encountered.
- **Email Updates:** Regular email updates to stakeholders on the status of the rollback.
- **Dedicated Slack Channel:** A dedicated Slack channel for real-time communication and issue resolution.

Post-Upgrade Monitoring and Maintenance

Following the CodeIgniter update/upgrade, Docupal Demo, LLC will provide ongoing monitoring and maintenance services to ensure system stability and optimal performance for ACME-1. Our team will closely monitor key metrics, including page load times, error rates, server resource utilization, and security alerts.

Monitoring Processes

We will proactively monitor ACME-1's updated CodeIgniter application. We will track and address any performance degradation or security vulnerabilities that arise.

Issue Tracking and Escalation

We will use Jira to track all identified issues. Critical issues will follow a predefined escalation path to ensure timely resolution. This includes immediate notification to designated ACME-1 contacts and prioritized attention from our senior engineering team.

Ongoing Maintenance

Sustained performance and security require regular maintenance. This includes:

- **Security Patching:** Applying security patches to the CodeIgniter framework and associated libraries.
- **Performance Monitoring:** Continuously monitoring application performance. We will identify and resolve bottlenecks.
- **CodeIgniter Version Updates:** Staying current with the latest CodeIgniter releases. We will apply updates and upgrades as needed.

Summary and Approval

This proposal outlines the plan to update or upgrade ACME-1's CodeIgniter framework. The primary benefits include enhanced security, improved performance, and easier maintenance. Potential risks involve compatibility issues, data migration challenges, and possible downtime during the process. Docupal Demo, LLC will work to mitigate these risks through careful planning and execution.

Approval Request

We request formal approval to proceed with this project. Following approval, the next steps are to:

1. Schedule a kickoff meeting with the ACME-1 team.
2. Finalize the development environment setup.
3. Begin the code migration process.

Signatures

We require signatures from both the CTO and the Head of Development at ACME-1 to authorize this project's commencement.