2024

FROM TECHNICAL RESOLUTION TO AGILE EVOLUTION: A BLENDED ROLE AT HUMANE INTERFACE DESIGN ENTERPRISE (HIDE)

Ketansingh Anil Patil
Michigan Technological University, kpatil4@mtu.edu

FROM TECHNICAL RESOLUTION TO AGILE EVOLUTION: A BLENDED ROLE AT HUMANE INTERFACE DESIGN ENTERPRISE (HIDE)

By

Ketansingh Anil Patil

A REPORT

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Computer Science

MICHIGAN TECHNOLOGICAL UNIVERSITY

2024

This report has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Computer Science.

Department of Computer Science

Report Advisor:	*Dr. Robert Pastel*
Committee Member:	*Dr. Leo Ureel*
Committee Member:	*Dr. Briana Bettin*
Department Chair:	*Dr. Zhenlin Wang*

Table of Contents

List of Figures

Acknowledgements

I express my gratitude towards Dr. Robert Pastel for his unwavering support in this project. I am also thankful to Michigan Technological University for providing me with the necessary resources to carry out the responsibilities successfully.

Abstract

This report provides a comprehensive overview of my journey through two distinct roles at the Humane Interface Design Enterprise (HIDE) - as a Technical Consultant in Spring 2023 and an Agile Process Analyst in Fall 2023. My tenure as a Technical Consultant was marked by the application of academic knowledge and technical skills to tackle real-world challenges at HIDE, leveraging experiences from my capstone projects and my roles in the industry as an IT Analyst Intern at Milwaukee Tool and a Data Engineer at UST. This role at HIDE enhanced my problem-solving skills and contributed to the progress of the project teams. Transitioning into the Agile Process Analyst role in Fall 2023 signaled a shift to a more observational and analytical approach within HIDE's software development processes, informed by my background in operational efficiency through technology at Milwaukee Tool. My engagement with HIDE was influenced by my interest in Software Development Engineering and Processes, particularly in system design and integration, process automation, and operational analysis. These areas, enriched by my industry experiences emphasizing Agile principles, and DevOps strategies, have steered my professional journey. The aim of this report is to articulate the evolution of my goals during my time at HIDE and highlight the skills I developed throughout this journey. It reflects on my contributions to HIDE's projects, underscoring the knowledge, and impactful experiences that have propelled my growth as a technical professional.

Introduction

In Spring 2023, my journey at HIDE began as a Technical Consultant, building upon a transformative summer internship at Milwaukee Tool. There, I honed my skills in process mapping, automation, and the development of manufacturing software, sparking a deep interest in application development, process observation, and analysis. Tasked with two problems, the role of Technical Consultant aimed to bridge the gap between academic learning and practical software development needs within HIDE. Before assuming the role, Dr. Pastel outlined a list of specific technical challenges faced by project teams, prompting me to apply my academic knowledge and industry experience to real-world problems. My focus was on solving the problem of deploying a Django application on the cPanel framework for the CS Scheduling App team and integrating the Microsoft Authentication Library (MSAL) for user-authentication purposes for the Little Brothers Friends of the Elderly (LBFE) team. These projects, detailed on the hidetech website, not only solved immediate issues but also served as documented use-cases that could guide future project teams at HIDE.

Transitioning to an Agile Process Analyst role in Fall 2023, I shifted focus from direct problem-solving to analyzing the LBFE team's software development process. This transition was catalyzed by my contributions in Spring, fostering a more observational and analytical approach. My efforts aimed at assessing Agile methodology application, teamwork, and communication effectiveness within LBFE, setting a precedent for process optimization. This audit of LBFE's software development process has the potential to inspire other teams at HIDE to adopt Agile practices, envisioning a common Agile framework that aligns with each team's unique challenges and objectives. This experience at HIDE not only enriched my technical skillset but also enhanced my understanding of Agile practices and the dynamics of team collaboration. The use-cases I developed, and the insights garnered from working with the project teams has the potential to benefit HIDE broadly, encouraging a culture of continuous improvement and collaborative success.

This report delves into these experiences, reflecting on the personal and professional growth I've achieved and the impact of my work on HIDE's project teams. Moreover, it invites readers to explore the intersections of technology, process analysis, and team dynamics within a unique academic and practical framework. Note that this study is not considered research since it is not generalizable. It is only applicable to HIDE. For definitions of specific terms mentioned in this report, please refer to Appendix A: Definitions.

Technical Consulting for HIDE (Spring 2023)

Responsibilities

The focus was on understanding and addressing challenges that were critical to HIDE's projects. My key duties included:

1. Solving problems encountered by HIDE teams which required researching and developing practical solutions in the form of use-cases.
2. Documenting the solutions in a way that everyone could understand, regardless of their technical background. This wasn't just about listing the steps, but also explaining the concepts behind them.
3. Collaboration with project teams and attending various meetings to promote a unified working environment as well as assisting team leads in navigating technical complexities.
4. Document the work on a WordPress website named ‘hidetech’ that I developed.

Problem 1: Deploying Django application on a cPanel Server

In Spring 2023, one of the HIDE teams encountered a challenge while working with Django, a web application framework. Although the team had experience in developing Django applications, they were new to deploying applications on cPanel, a common web hosting platform. This knowledge gap in moving from development to deployment became an obstacle in completing the project.

Implementation

The approach was twofold: first, to develop a working proof-of-concept for deploying on cPanel and second, to document the process in a step-by-step manner.

1. **Setting Up a Django Environment**: This included configuring the Django application to suit the cPanel environment.
2. **WSGI Configuration**: Understanding the Web Server Gateway Interface (WSGI) and configuring it correctly.
3. **Deployment Testing**: Continuous testing followed the deployment to ensure functionality and performance were intact.

Documentation

The process was documented on the ‘hidetech’ website. It included:

1. An introduction to Django and its features.
2. A general procedure for deploying Django applications.
3. A step-by-step tutorial on deploying the application on cPanel, emphasizing concepts such as WSGI vs ASGI, application object configuration, and Django environment settings.

Problem 2: User Authentication with Microsoft Authentication Library (MSAL) and Postman: Acquiring a Bearer Token

My second deliverable as a Technical Consultant involved integrating the Microsoft Authentication Library (MSAL) for one of the modules in the Little Brothers Friends of the Elderly (LBFE) project. This integration was important to enhance the security and streamline the process of user authentication in the project.

Understanding MSAL

MSAL is a library used to authenticate users and applications to Azure Active Directory (Azure AD), Microsoft Identity Platform, and Microsoft Accounts. It supports various authentication methods, including OAuth 2.0 and OpenID Connect, making it a secure choice for modern applications.

Implementation

The process involved several key steps:

1. **Registration with Azure AD**: The initial phase involved registering the application with Azure AD. This step was crucial in defining the parameters for the application's interaction with Azure AD for authentication purposes.
2. **Configuring Authentication Flows**: Given the varied types of applications that could interact with the system, configuring the right authentication flow was critical. This involved understanding OAuth 2.0 flows, including authorization code grant and implicit grant flows, depending on the application type.
3. **Token Acquisition and Management**: A core part of the integration was handling token acquisition. The process required setting up mechanisms to acquire, refresh, and manage access tokens, which are essential for authorizing user access.
4. **Demo**: To ensure the effectiveness of the integration, a demo was set up. This demo included manually running the token acquisition process using Postman.

Documentation

The documentation covered:

1. Introduction to MSAL
2. Steps for Registering an Application with Azure AD
3. Authentication Flow Concepts
4. Bearer Token Usage
5. Demo

Hidetech Website

Documentation was a crucial aspect of the role, emphasizing the accessibility and clarity of technical information. I developed a website called ‘hidetech’ to document my deliverables. The website was developed to be educational, ensuring that even complex technical details were easily understandable for all.

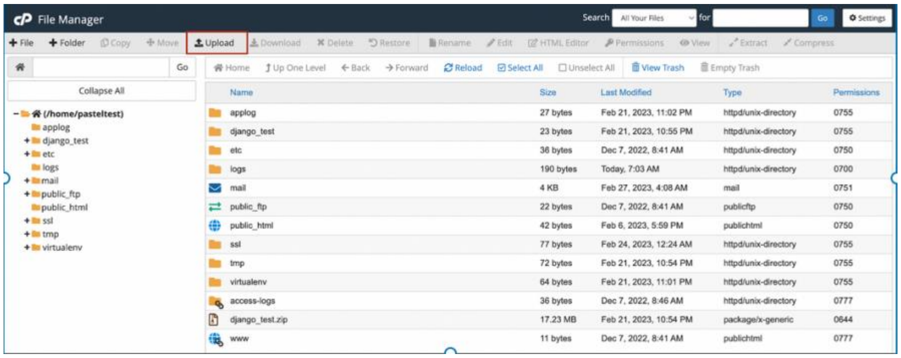
Accessibility

One of the key focuses of the site was ‘Web Accessibility’. This meant designing the site with features that make it usable by all people, regardless of their abilities or disabilities.

The site includes:

1. Text alternatives for non-text content. For example,

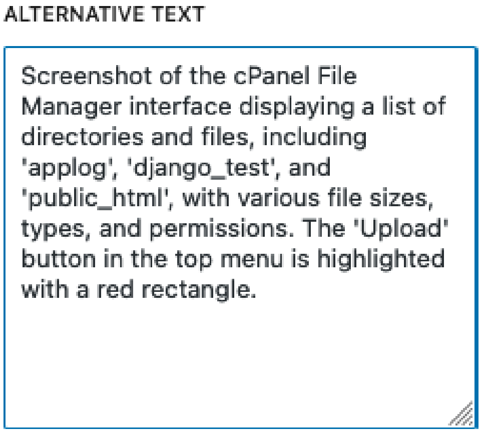
   Image:



Name	Size	Last Modified	Type	Permissions
applog	27 bytes	Feb 21, 2023, 11:02 PM	httpd/unix-directory	0755
django_test	23 bytes	Feb 21, 2023, 10:55 PM	httpd/unix-directory	0755
etc	36 bytes	Dec 7, 2022, 8:41 AM	httpd/unix-directory	0750
logs	190 bytes	Today, 7:03 AM	httpd/unix-directory	0700
mail	4 KB	Feb 27, 2023, 4:08 AM	mail	0751
public_ftp	22 bytes	Dec 7, 2022, 8:41 AM	publicftp	0750
public_html	42 bytes	Feb 6, 2023, 5:59 PM	publichtml	0750
ssl	77 bytes	Feb 24, 2023, 12:24 AM	httpd/unix-directory	0755
tmp	72 bytes	Feb 21, 2023, 10:54 PM	httpd/unix-directory	0755
virtualenv	64 bytes	Feb 21, 2023, 11:01 PM	httpd/unix-directory	0755
access-logs	36 bytes	Dec 7, 2022, 8:46 AM	httpd/unix-directory	0777
django_test.zip	17.23 MB	Feb 21, 2023, 10:54 PM	package/x-generic	0644
www	11 bytes	Dec 7, 2022, 8:41 AM	publichtml	0777

Figure 1: cPanel File Manager

   Alt Text:

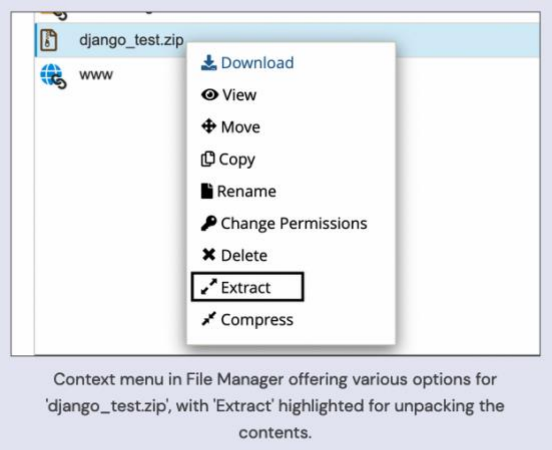


ALTERNATIVE TEXT

Screenshot of the cPanel File Manager interface displaying a list of directories and files, including 'applog', 'django_test', and 'public_html', with various file sizes, types, and permissions. The 'Upload' button in the top menu is highlighted with a red rectangle.

Figure 2: Alt Text for cPanel File Manager Image

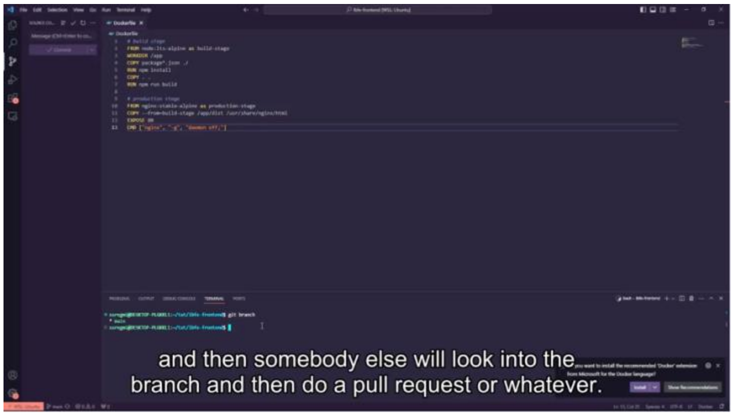
2. Image Caption.



Context menu in File Manager offering various options for 'django_test.zip', with 'Extract' highlighted for unpacking the contents.

Figure 3: Image Caption

3. Video clip with accurate subtitles.



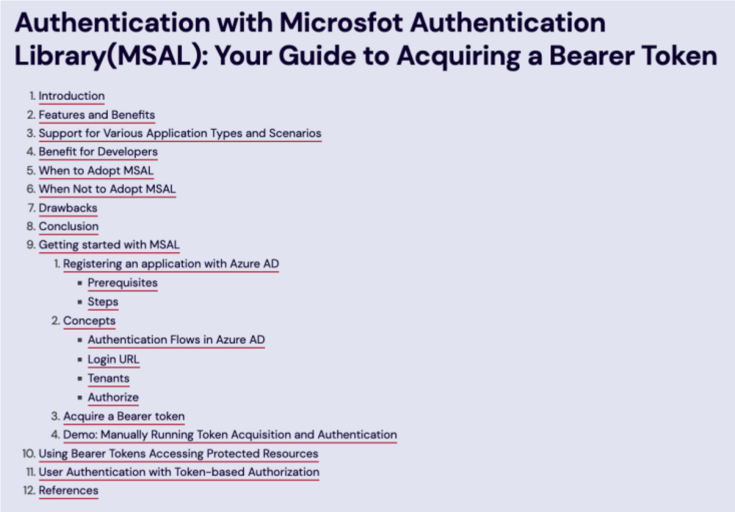
```
# build stage
FROM node:lts-alpine as build-stage
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build

# production stage
FROM nginx:stable-alpine as production-stage
COPY --from=build-stage /app/dist /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

```
$ git branch
* main
$
```

and then somebody else will look into the branch and then do a pull request or whatever.

Figure 4: Git Video Clip Subtitles

4. Easy navigation and identification of content.



Authentication with Microsfot Authentication Library(MSAL): Your Guide to Acquiring a Bearer Token

1. Introduction
2. Features and Benefits
3. Support for Various Application Types and Scenarios
4. Benefit for Developers
5. When to Adopt MSAL
6. When Not to Adopt MSAL
7. Drawbacks
8. Conclusion
9. Getting started with MSAL
    1. Registering an application with Azure AD
        - Prerequisites
        - Steps
    2. Concepts
        - Authentication Flows in Azure AD
        - Login URL
        - Tenants
        - Authorize
    3. Acquire a Bearer token
    4. Demo: Manually Running Token Acquisition and Authentication
10. Using Bearer Tokens Accessing Protected Resources
11. User Authentication with Token-based Authorization
12. References

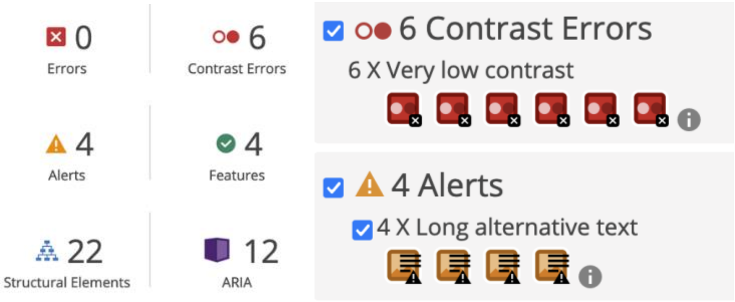
Figure 5: MSAL Page Index

Evaluation and Areas of Improvement

To evaluate the accessibility of the 'hidetech' website, I utilized the WAVE Web Accessibility Evaluation Tool. This tool is used to identify accessibility concerns and ensuring content is accessible to everyone.

After conducting an analysis on four of webpages, here are the metrics:

1. **Homepage**



0 Errors

6 Contrast Errors

4 Alerts

4 Features

22 Structural Elements

12 ARIA

☑ 6 Contrast Errors

6 X Very low contrast

☑ 4 Alerts

☑ 4 X Long alternative text

Figure 6: Homepage Accessibility Metrics

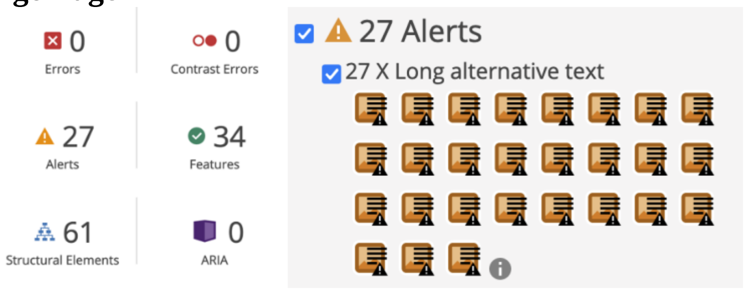
Evaluation

The WAVE tool's evaluation of the webpage indicates aspects of accessibility with 0 errors detected, ensuring a good foundation for web content accessibility standards. The presence of 4 features suggests that there are positive aspects of accessibility incorporated into the page. The 22 structural elements found imply that the page has a well-defined structure, which can aid in navigation and comprehension for users utilizing assistive technologies.

Areas of Improvement

There are opportunities for enhancement, particularly in addressing the 6 contrast errors that could hinder readability for users with visual impairments. Rectifying these will involve adjusting text and background colors to increase contrast ratios. Furthermore, the 4 alerts for long alternative text indicate that while descriptive images are good for accessibility, overly verbose descriptions may overwhelm screen readers, and these should be made more concise.

2. Deploy Django Page


0 Errors

0 Contrast Errors

27 Alerts

34 Features

61 Structural Elements

0 ARIA

☑ 27 Alerts

☑ 27 X Long alternative text

Figure 7: Django Page Accessibility Metrics

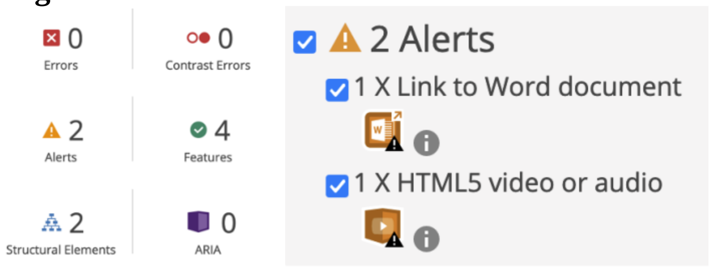
Evaluation

The accessibility assessment of the webpage conducted revealed a well-structured webpage with 34 accessibility features and 61 structural elements, indicating a solid framework for users and assistive technologies. No errors or contrast issues were detected, showcasing an adherence to accessibility guidelines.

Areas of Improvement

The tool flagged 27 alerts for long alternative text, suggesting that while images are descriptively tagged, these descriptions may be verbose.

3. Git Tutorial Page


0 Errors

0 Contrast Errors

2 Alerts

4 Features

2 Structural Elements

0 ARIA

☑ 2 Alerts

☑ 1 X Link to Word document

☑ 1 X HTML5 video or audio

Figure 8: Git Tutorial Page Accessibility Metrics

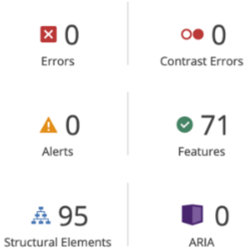
Evaluation

The accessibility assessment of the webpage has yielded positive results, with 0 detected errors or contrast issues, indicating a solid foundation in accessibility. The presence of 4 features demonstrates active measures taken to ensure user accessibility, supported by 2 structural elements that provide navigational ease.

Areas of Improvement

Despite these results, the evaluation brought attention to 2 alerts that warrant further action. One is for a link to a Word document and another for an HTML5 video or audio element. Although a downloadable link to transcripts is provided below the Git video on the webpage, to elevate the page's accessibility, providing downloadable alternative formats for documents and transcripts or captions for multimedia elements is still recommended.

4. **MSAL Page**



0 Errors

0 Contrast Errors

0 Alerts

71 Features

95 Structural Elements

0 ARIA

Figure 9: MSAL Page Accessibility Metrics

Evaluation

The webpage is significantly better in terms of accessibility, with the WAVE tool reporting 0 errors, 0 contrast errors, and an absence of alerts. A total of 71 features are recognized for enhancing accessibility, alongside 95 structural elements that contribute to a well-organized and navigable site. The absence of ARIA (Accessible Rich Internet Applications) errors suggests that the page is well-equipped for assistive technologies.

Areas of Improvement

Maintaining this level of accessibility requires consistent monitoring, especially as new content is added, or existing content is updated. Regular audits with WAVE and other related tools will help ensure adherence to Web Content Accessibility Guidelines (WCAG) and a seamless user experience for all users.

Experiential Learning

Experiential learning on the ‘hidetech’ website embodies a hands-on approach, transforming theoretical knowledge into practical skills through engagement. By offering step-by-step tutorials, video clip on version control or Git which is one of the complex steps in the software development process, concise concept explanations, scenarios of real-world applications, illustrative code snippets, and technical resources to upskill, the platform ensures that technical content is accessible and comprehensible. This method not only facilitates learning by doing but also encourages learners to apply concepts in practical contexts, enhancing their understanding and retention of technical information.

The documentation provided on ‘hidetech’ is segregated into the following sections:

1. Step-by-step tutorials
2. Video tutorial on Git
3. Concept explanations
4. Real-world application scenarios
5. Code snippets and screenshots for clarity
6. Role wise technical resources to learn new tools and technologies

Reflection

My time as a Technical Consultant at HIDE was a valuable experience where I learned about application deployment and solving real world industry challenges. This role bridged the gap between academic theories and their real-world application in software development, offering a practical perspective on my work. In addition, overcoming technical challenges not only improved my skills but also sharpened my problem-solving abilities. Looking forward, my experience as a Technical Consultant has equipped me with the tools and confidence to tackle future challenges in technology.

Observing and Documenting the Software Development Process (Fall 2023)

Responsibilities

In the Fall of 2023, I worked as an Agile Process Analyst for the LBFE (Little Brothers - Friends of the Elderly) team at HIDE. This deliverable involved understanding and assessing how effective the current software development process implemented in projects at HIDE are, identifying any obstacles that slowed down the progress, and suggesting possible approaches to mitigate them. I was tasked on working with the LBFE team, particularly, observing, and reporting how they applied Agile methods, managed the project, and tackled challenges throughout the Software Development Lifecycle (SDLC). This involved participating in meetings, brainstorming with the team lead, and analyzing different stages like user story development, task allocation, teamwork, and stakeholder communication.

What is an Agile Process Analyst

An Agile Process Analyst specializes in auditing and analyzing the Agile process within a software development team. They focus on understanding how Agile methodologies are being implemented, identifying areas for improvement, and ensuring that Agile practices are effectively implemented for project success.

The LBFE project

The LBFE project at HIDE is an initiative under the umbrella of Little Brothers - Friends of the Elderly or LBFE, a network of non-profit organizations committed to providing essential services to the elderly. The project team was tasked with developing an application that aims to streamline the process of scheduling and managing transportation services, ensuring efficient utilization of vehicles and volunteers.

Software Development Process at LBFE

The choice of using Agile software development process by the team lead was influenced by its suitability for the team's educational context, prior experience of the team lead at LBFE in the previous semester and their experiences in the form of internships. The role of ‘Project Manager’ or ‘Team Lead’ as per HIDE’s nomenclature is a role with a mix of organizational, managerial, and technical responsibilities. The development team with completely technical responsibilities was adapted to fit the overall team’s academic commitments and learning objectives, deviating from traditional Agile roles seen in professional/industry environments where the Project Manager solely focuses on timely completion of project deliverables, and budgeting.

Agile Manifesto Principles

In the project, Agile’s core values were embraced by focusing on people, teamwork, and being open to change. Regular team meetings were held, and open lines of communication were maintained, with a priority placed on personal interaction over strict procedures. The team lead played a key role in nurturing this cooperative spirit, ensuring the approach stayed true to the importance Agile places on personal interactions and teamwork.

User Stories

Close collaboration with the client led to the creation of user stories aimed at fulfilling client requirements. For example, when the client needed a monthly overview of transportation services for cost analysis, the team converted this request into a series of tasks or stories. Each user story was designed to be completed within 2 to 4 weeks and was either an independent task or part of a larger user story. One specific task involved breaking down the client's request into monthly phases to develop a page meeting their requirements. Additionally, ambiguities in user stories were carefully addressed to prevent misunderstandings. A key issue was defining the requirements of the 'Updates' page of the application, which took 3 meetings to clarify. After clarification, tasks were prioritized according to their importance to the client. Task assignment involved teaming up members to work on different parts of the user story, either on the frontend or backend.

Team Meetings

Bi-weekly meetings played a pivotal role in steering both the project and team dynamics. These sessions not only facilitated tutorials and learning opportunities led by the team lead but also incorporated pair programming to bolster skill development, with a focus on database management. Enhanced communication and transparency were direct outcomes of these gatherings, which covered discussions on backlogs, bug fixes, current challenges, and future planning. The utilization of Trello and Discord for project updates and ongoing communication, alongside the team leader’s bi-weekly one-on-ones for task management and mentorship, ensured a perfect alignment of tasks with individual members' skills and technological knowledge. This setup fostered conflict resolution and boosted team morale, setting an emphasis on skill development and personalized mentorship within the team structure.

Pair Programming

The strategy behind pair programming was largely influenced by team members' interests and the specific nature of the work required. The team lead assigned frontend tasks based on expressed interest, providing a list of objectives to those inclined towards frontend development. Conversely, a focus on backend tasks was mandated for the remainder of the team, addressing previous semesters' neglect. Communication for most pair programming sessions was facilitated through Discord, ensuring efficient collaboration and alignment with project goals for both frontend and backend tasks.

Client Interaction

The team lead focused on engaging the client through bi-weekly presentations to review progress and gather feedback. This approach ensures the project meets client expectations and responds to their needs. Regular meetings, which started virtually and then shifted to in-person meetings at LBFE's office in Hancock, MI, enhance collaboration and concentrate on improving the client's work processes. When new features are proposed by the client, the team lead evaluates their impact on current priorities, showing adaptability. This dedication to client-focused development helps shape the application according to the client's changing needs, indicating a strategic and flexible approach to managing the project.

Testing

The project emphasized developing through consistent feedback and client-involved testing. While initially focused on creating prototypes for the client to test, the approach evolved to include the client as a primary tester, enabling iterative product testing. The team lead played a crucial role, reviewing code changes and providing targeted feedback to developers for necessary enhancements, ensuring the application aligns with project goals and stakeholder expectations.

CI/CD

The LBFE project is exploring Continuous Integration/Continuous Deployment (CI/CD) using Jenkins or GitHub Actions, aiming for deployment on the LBFE server. Currently, code integration and testing are conducted manually, including Git review and local functionality tests. Team members work on individual features and engage in pair programming, each operating on remote branches. Upon feature completion, the team lead performs manual testing on the branch, addressing any bugs with responsible members or involving other pairs as needed to maintain project momentum and adherence to timelines.

Challenges and Learnings

The project faced challenges in balancing project commitments with academic duties. Time management emerged as a critical issue, with team members juggling their coursework, extracurriculars, and project responsibilities. Furthermore, the 1 credit value of the Enterprise (ENT) course, which facilitated the project, offered minimal incentive for substantial contribution. Additionally, navigating a large codebase and limited experience with technologies like Vue.js and JavaScript presented difficulties, compounded by the steep learning curve associated with these tools.

On the contrary, the team learned valuable lessons in software development. A significant insight came from an issue where assigning a backend task to a member inexperienced with databases caused disruption. This highlighted the need to match tasks with the right experience and conduct thorough code reviews. In terms of achievements, the team in the previous semester completed ~80% of the frontend tasks, mainly due to pair programming. This method improved problem-solving and teamwork. Its success inspired the current team to adopt the same approach, aiming to overcome frontend development challenges like their predecessors. This strategy now guides the team's current efforts.

Good Practices and Insights

The project utilized Trello for task management, enhancing accountability and progress visibility. Pair programming improved team dynamics, fostering better collaboration and coding confidence. Communication was managed through Discord, supporting both group and pair interactions. The team lead provided technical skill development through tutorials, focusing on Git, Express.js, and other technologies. Bi-weekly one-on-one meetings were key to aligning individual goals with team objectives. Despite being new to software development, the team displayed great work ethic and a willingness to learn, overcoming initial inexperience through hands-on exercises and mini projects. The team lead noted their eagerness and responsibility-taking as major strengths, viewing their lack of experience as an opportunity for growth, despite concerns about potential disheartenment from early challenges.

Project Closure

In the final phase of the project, the team’s objective is to complete 4 key web pages of the LBFE application. The project is expected to extend over 2 more semesters, largely due to the existing codebase and learning curve for new members inexperienced in web technologies, making it challenging to predict a precise completion date. Initially estimated to take 6 semesters, this timeline indicates the depth and breadth of work required to fulfill LBFE's needs.

Conclusion

In the project, the team leader set realistic goals considering its role in a 1-credit course. Acknowledging the students' limited time, an hour outside class was recommended for project tasks. The leader's role extended beyond coding to facilitate task progression and supporting the team, especially important for those new to development. Weekly meetings served as a platform for collaboration, learning, and progress tracking. Overall, the lead effectively balanced project management with educational needs, focusing on clear expectations and fostering student development within LBFE in software engineering.

Process Enhancement Strategies

This section of the document outlines strategies for enhancing the development process through different approaches. It proposes integrating industry-standard tools and practices to align educational project work with professional software development standards, aiming to equip students with real-world skills and improve project outcomes.

Here’s a table summarizing the difference between the Agile at LBFE to Traditional Industry Agile:

Aspect	**LBFE Project**	**Traditional Industry Agile**
Flexibility and Adaptability	High, tailored to academic schedules	Moderate, market-driven
Incremental Development	Iterative, longer phases	Rapid, shorter iterations
Client Engagement	Direct and integrated	Varies, less intensive
Collaboration and Communication	Mentorship-focused, learning-oriented	Efficiency-focused, role-specific
Rapid Delivery	Progressive, educationally constrained	Rapid, market-oriented
Risk Management	Adapted to academic context	Market and technology-focused
Quality Focus	Iterative testing, client feedback	Automated, streamlined testing
Empowerment and Self-Organization	High, with educational focus	High, within market scope
Feedback and Improvement	Client and internal retrospectives	Market and technology feedback
Modern Practices	Adapted to educational setting	Closely aligned with industry trends
Scalability	Focused on project and education	Focused on market and expansion

Simplicity and Clarity	Tailored for learning curve	Aimed at market efficiency
Project Suitability	Educational projects	Broad industry application
Past Successes	Influenced by educational experiences	Driven by market trends
Project Management Tools Used	Trello, Discord	JIRA, Confluence, Slack

To ensure the completion and delivery of the LBFE project within the timeframe of 2 semesters, it is crucial to address specific areas within the Agile process that require improvement. This necessitates the identification of gaps in the current methodology and the proposal of actionable strategies. The potential approaches outlined below aim to bridge these gaps, enhancing the overall effectiveness of the project management process.

I. Incremental Development and Rapid Delivery Alignment

The LBFE project's longer iterative phases risk delaying final delivery. Adopting the "Scrum of Scrums" approach, which organizes the team into smaller, cross-functional groups each responsible for a project module, could mitigate this. These groups conduct daily stand-ups to address impediments swiftly, with weekly meetings among representatives to ensure alignment and integration. This method, inspired by large enterprise Agile projects, but condensed to accommodate and address the issues of a student led project aims to balance educational objectives with the need for rapid, industry-standard delivery.

This is how the Scrum of Scrums approach is implemented:

1. **Division into Cross-Functional Teams**: Segment the team into smaller units focused on specific project components. These self-organizing teams encompass all necessary skills for product increment delivery.
2. **Regular Scrum Meetings**: Units hold daily meetings to discuss progress and plan, facilitating swift communication and decision-making.
3. **Scrum of Scrums Meeting**: Representatives from each unit convene in a broader meeting 2 to 3 times a week to discuss integration and dependencies, ensuring project coherence.
4. **Problem-Solving and Adaptation**: When issues are identified that impact more than one unit, collective problem-solving is necessary. This forum addresses multi-unit issues with collective brainstorming and plan adjustments to keep the project on track.
5. **Feedback Loop**: Ensures decisions from the Scrum of Scrums are communicated back to individual units, maintaining clarity and direction across the project.

This streamlined approach is designed to enhance coordination and accelerate project delivery while taking into consideration the educational context.

Example

Consider a project where one unit is responsible for the user interface, another for the database, and a third for the application logic. During the Scrum of Scrums, the representatives realize that a change in the database schema is necessary, which will impact both the user interface and application logic teams. They collectively decide on a new schema that supports the needs of all units and plan a coordinated update to their respective parts of the project.

If Scrum of Scrums is implemented in the LBFE project, it would have a clear focus on educational objectives while also mirroring the rapid iteration and integration seen in professional Agile environments.

II. Quality Assurance (QA)

Quality Assurance is a critical component in software development, ensuring that the product not only meets the functional requirements but also provides a seamless and bug-free user experience. For the team, integrating QA practices into their workflow is essential for delivering a robust software.

This is how quality assurance can be achieved:

1. **Understanding the QA Process**: The first step is to conduct workshops focusing on the importance of QA in the development lifecycle. Students need to understand different testing methodologies like unit testing, integration testing, system testing, and acceptance testing.
2. **Writing Test Cases**: Students should be encouraged to write test cases alongside their code. This practice ensures that they think about the different scenarios in which their code will be used and helps them understand the importance of covering edge cases.
3. **Automated Testing**: Introducing students to automated testing tools such as Selenium for web applications or JUnit for Java applications can significantly enhance their testing capabilities. These tools can run a suite of tests automatically, saving time and ensuring consistency.
4. **Continuous Integration/Continuous Deployment (CI/CD)**: Guide the students through setting up a CI/CD or to be precise, a Jenkins pipeline. This will involve installing Jenkins, integrating it with their source code repository (like GitHub or Bitbucket), and configuring it to perform builds with every code commit.

5. **Automated Builds**: Each time code is committed, Jenkins can automatically compile the code and run tests. This immediate feedback is vital for identifying issues early in the development cycle.
6. **Deployment Automation**: Once the build and tests are successful, Jenkins can deploy the code to a staging environment. This could be a server on campus or a cloud service like AWS or Heroku. The staging environment closely replicates the production setup, allowing for realistic testing.
7. **Client Feedback**: Integrate a feedback loop with the client. After deploying to the staging environment, invite stakeholders to review the application.
8. **Retrospectives**: After each iteration, hold a retrospective meeting where student developers can discuss what went well and what didn't in terms of QA. This meeting should result in actionable items that aim to improve the QA process in the next iteration.

Example

Imagine the team is working on a web application. One team member is adding a new feature that requires changes to the backend API. They commit their code, and the CI/CD pipeline is triggered. Here's the process:

1. **Build:** Jenkins automatically fetches the latest code and starts building the project with the new changes.
2. **Test:** Once the build is successful, Jenkins runs a suite of automated tests, including new tests written for the API changes.
3. **Results:** The tests reveal a bug where the new API changes are incompatible with the existing frontend code, causing several test cases to fail.
4. **Feedback:** Jenkins sends an alert to the team's communication platform, indicating which tests failed and why.
5. **Action:** The team reviews the feedback, and the developer responsible for the API changes works on fixing the code. They push the new changes, and the CI/CD process starts again.
6. **Deployment:** After the fix, all tests pass, and Jenkins automatically deploys the updated backend to the staging environment for further acceptance testing.

This example demonstrates how a CI/CD pipeline facilitates a consistent QA process and provides immediate feedback to developers, allowing for quick resolution of issues. By integrating these QA and feedback mechanisms into their project, students will not only be able to ensure the quality of their software but also gain practical skills in modern software development practices.

III. Risk Management Strategy

In the context of the LBFE project, where student developers are at the helm, an effective risk management strategy needs to be straightforward yet comprehensive. A simplified approach is to utilize a "Bug/Risk Register" — a basic, living document that records potential bugs, risks, their impact, likelihood, and mitigation strategies.

Here's how they can achieve this and its benefits:

1. **Implementation**: Students can be guided to create a Bug/Risk Register in a shared spreadsheet. The document should have columns for bug/risk description, impact level (high, medium, low), likelihood (high, medium, low), mitigation actions, and responsible team members. This register should be reviewed and updated during bi-weekly meetings.
2. **Benefits**: This method teaches students to continuously monitor for new risks and reassess their impact and likelihood as the project progresses. It also fosters a proactive mindset, encouraging them to think ahead about potential problems and how to address them.

Example

A use-case might involve a project team that is integrating third-party payment processing into their application. The bugs/risks involved could include API changes, service outages, or security vulnerabilities. In their Bug/Risk Register, they would note, rate its impact and likelihood, and detail mitigation steps, such as implementing testing, debugging, monitoring service status, or scheduling regular security reviews.

By incorporating a Bug/Risk Register into their project workflow, student developers learn a vital aspect of software project management.

IV. Leveraging Cloud's Capabilities

The LBFE project lacked application deployment capabilities and heavily relied on manual testing. The project could leverage cloud computing which offers a versatile platform for developers to deploy, manage, and scale their applications efficiently. The inherent flexibility and scalability of cloud services make them ideal for projects, especially considering the varying demands and limited resources often associated with student-led projects.

Here are some of the best aspects of adopting cloud:

1. **Deployment Ease and Cost-Effectiveness**: Cloud platforms like AWS (Amazon Web Services), Microsoft Azure, and Google Cloud offer free tiers, which are ideal for students. These free tiers usually include enough resources to host small to medium-sized applications, making them cost-effective for educational projects.

They should, however, be mindful of the limitations and pricing structures of these services to avoid unexpected charges. Using cloud services, student developers can deploy their applications without the need for physical hardware or extensive setup, reducing the initial investment costs significantly.

2. **Resource Allocation and Elasticity**: One of the key benefits of cloud computing is the ability to allocate resources dynamically. This means that student developers can start with minimal resources and scale up as the demand for their application grows. This elasticity is crucial for managing costs effectively, as students only pay for the resources they use.
3. **Scalability and Performance**: Cloud platforms are designed to handle varying loads, making them ideal for projects that may experience fluctuating levels of user engagement. This scalability ensures that the application remains stable and performs well, even as the number of users increases or during peak usage times.

Example

Consider a project which includes developing a web application for a university event. Initially, the application might only have a few users, mainly the organizing team. As the event date approaches, the user base could expand, including students, faculty, and external visitors. By hosting the application on a cloud platform like AWS, the team can start with a small server instance and a basic relational database under the free tier. As the user base grows, they can scale up the resources, perhaps moving to a larger server instance or enabling auto-scaling to adjust resources automatically based on traffic. This approach ensures the application remains responsive and stable throughout the event, despite the fluctuating demand.

In summary, cloud computing offers a practical and cost-effective solution for student developers, allowing them to deploy and manage applications with ease. The key is to start with the free tiers offered by cloud providers, closely monitor resource usage, and scale up as necessary, keeping an eye on potential costs.

Conclusion

In conclusion, enhancing the LBFE project by integrating practices such as the Scrum of Scrums for coordinated development, Quality Assurance through CI/CD pipelines, and practical cloud computing applications for scalability and deployment is vital for aligning with industry standards and rapid delivery. This approach not only ensures the successful completion of the project but also equips student developers with the essential skills and understanding of cost considerations in technology, preparing them for real-world challenges.

Software Development Process Evaluation

The insights of the LBFE team, as crucial contributors to the project's development, are essential for guiding future strategies and methodologies. This survey aims to gather feedback on different facets of the project, such as team dynamics, the efficacy of communication, and individual learning experiences. These findings will be integral in driving the collective improvement and success of the LBFE team. In addition, the gathered responses are anticipated to play a role in bettering the software development process within HIDE teams.

Quantitative Analysis

The quantitative evaluation provides a view of the LBFE team's engagement with the project, examining several key factors. The assessment was structured through a survey that included both a 1 to 5 rating scale and numerical data regarding hours dedicated by the team per week for the project. The 1 to 5 scale is a Likert scale where 1 represents *'Very Ineffective*,' escalating to 5 which stands for *'Very Effective*.' Each number on the scale is defined as follows: 1 indicates a strong negative sentiment, 2 suggests modest effectiveness needing improvement, 3 represents neutrality, 4 signals positive effectiveness with minor issues, and 5 denotes exceptional performance without significant drawbacks. This scale was selected for its clarity and interpretative ease, allowing team members to articulate their views across diverse aspects of the project systematically. Such a scale also provides valuable nuance, capturing varying levels of satisfaction or dissatisfaction, from potentially highlighting areas requiring further training and support (scores of 1 or 2) to demonstrating areas of strong performance and skill acquisition (scores of 4 or 5).

The responses, collected from a limited sample size of 5, present challenges due to the potential for limited variability. Nonetheless, the collected data is foundational in understanding the project's multifaceted nature. In interpreting this compact dataset, both mean and median were utilized to identify central tendencies, with the application of each measure dependent on the specific question's data distribution.

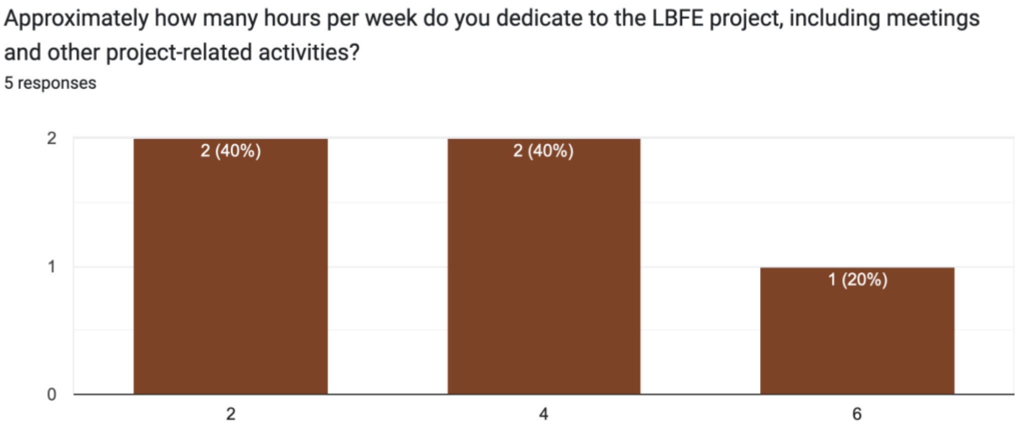
Following this introduction to the methods and scales used in our quantitative analysis, we will delve into the specifics of the following metrics:

1. Hours Dedicated per Week
2. Balance Between Academic Responsibilities and Project Work
3. Team Collaboration and Dynamics
4. Technical Skill Level at the Start versus the End of the Fall 2023 Semester
5. Effectiveness of Resources and Tools
6. Effectiveness of Project Documentation.

These metrics have been chosen to offer a comprehensive evaluation, presenting a multifaceted view of the LBFE project's impact on the software development process and the team's growth over the semester.

The subsequent sections will outline the rationale for choosing either the mean or median for each question, illustrating how these measures elucidate the collective assessments and individual experiences within the project's scope. This analysis is aimed at providing a detailed portrayal of the team's perspective on their participation and the effectiveness of the support structures throughout the LBFE project.

Metric 1: Hours Dedicated per Week


Approximately how many hours per week do you dedicate to the LBFE project, including meetings and other project-related activities?

5 responses

Figure 10: Hours Dedicated Per Week

- Results
  - 2 hours: 2 responses (40%)
  - 4 hours: 2 responses (40%)
  - 6 hours: 1 response (20%)
- Mean (Average): 3.6 hours.
- Median: 4 hours.

For metric 1, with no outliers and responses evenly spread, the mean effectively depicts the overall average time dedication to the project, accounting for the total hours contributed by all respondents. This makes it ideal for representing the collective effort. The median, though useful for identifying the most common experience, may not fully capture the variation in hours dedicated by all team members. Hence, the mean is preferred for a summary of time allocation to the project.

Metric 2: Balance Between Academic Responsibilities and Project Work

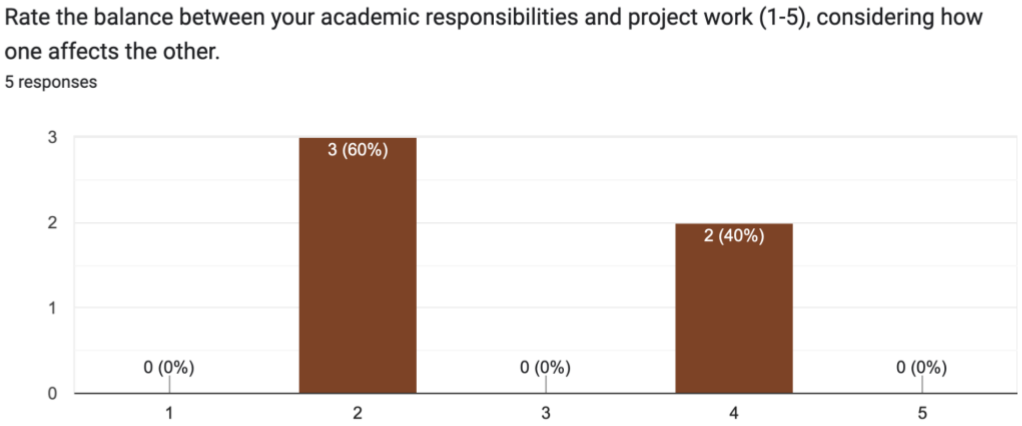


Figure 11: Balance Between Academic and Projects

- Results
  - Rating 1: 0 responses (0%)
  - Rating 2: 3 responses (60%)
  - Rating 3: 0 responses (0%)
  - Rating 4: 2 responses (40%)
  - Rating 5: 0 responses (0%)
- Mean (Average): 2.8
- Median: 2

For the assessment of balance between academic and project work, the median is more suitable as it best represents the typical response in an ordinal dataset, accurately reflecting the most common rating without assuming equal intervals between the ratings. This measure of central tendency is crucial for understanding participants' experiences, especially in areas like balancing academic responsibilities with project work.

Metric 3: Team Collaboration and Dynamics

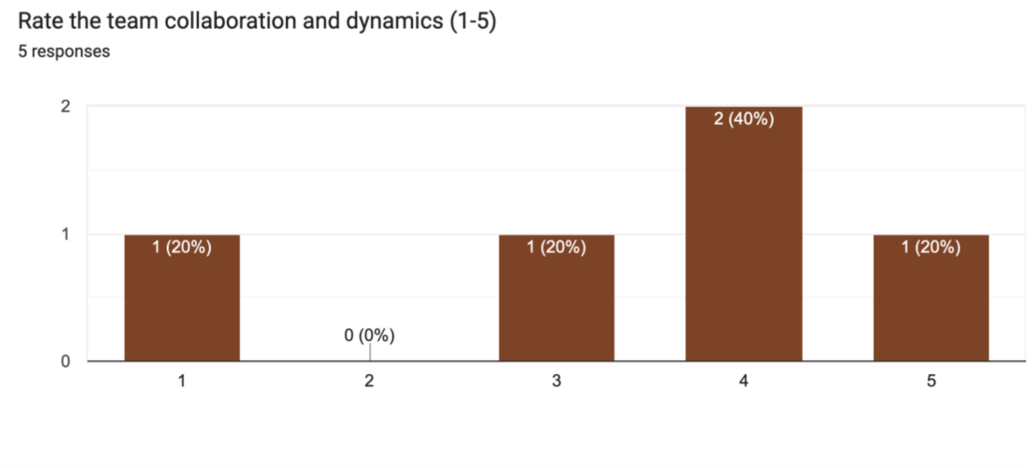


Figure 12: Team Collaboration and Dynamics

- Results
  - Rating 1: 1 response (20%)
  - Rating 2: 0 responses (0%)
  - Rating 3: 1 response (20%)
  - Rating 4: 2 responses (40%)
  - Rating 5: 1 response (20%)
- Mean (Average): 3.2
- Median: 3

For evaluating metric 3, where ratings are spread across the scale, the median provides a clearer representation of the central experience among team members. It offers an insightful snapshot of the team's view on collaboration and dynamics, unaffected by the spread of responses and avoiding the potential distortion an average might introduce. Therefore, the median is the preferred measure to convey the most typical team experience regarding collaboration and dynamics in the LBFE project.

Metric 4: Technical Skill Level (Start vs. End of Semester)

Semester Start

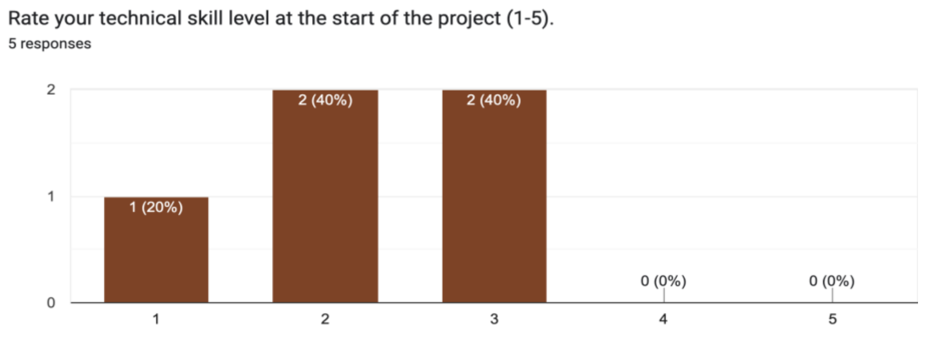


Figure 13: Skill level at the start of the semester

Semester End

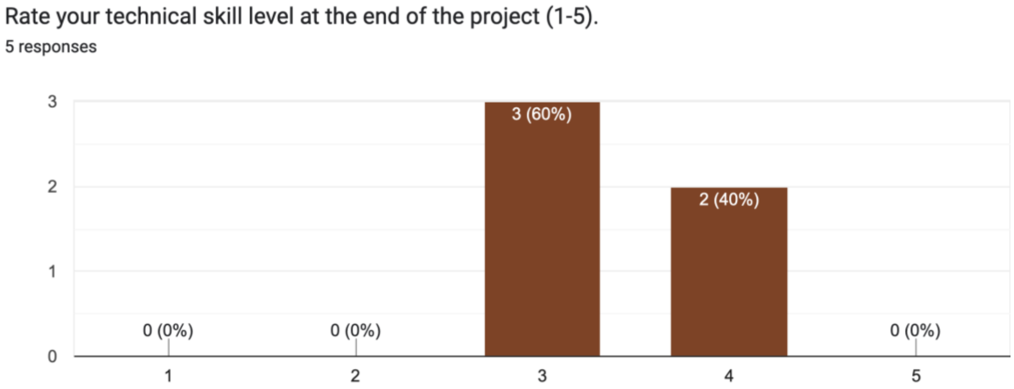


Figure 14: Skill level at the end of the semester

Results

- Start of the semester
  - Rating 1: 1 response (20%)
  - Rating 2: 2 responses (40%)
  - Rating 3: 2 responses (40%)
  - Rating 4: 0 responses (0%)
  - Rating 5: 0 responses (0%)
- End of the semester
  - Rating 1: 0 responses (0%)
  - Rating 2: 0 responses (0%)
  - Rating 3: 3 responses (60%)
  - Rating 4: 2 responses (40%)
  - Rating 5: 0 responses (0%)
- Start of the Project (Start of the Fall 2023 semester)
  - Mean (Average) - 2.2
  - Median - 2
- End of the Project (End of the Fall 2023 semester)
  - Mean (Average) - 3.4
  - Median - 3

The median is appropriate both at the start and end of the project because it represents the most common rating category without being skewed by extreme values. It indicates the most typical skill level among participants, allowing us to identify the prevalent experience regarding technical skill improvement throughout the semester.

Metric 5: Effectiveness of Resources and Tools

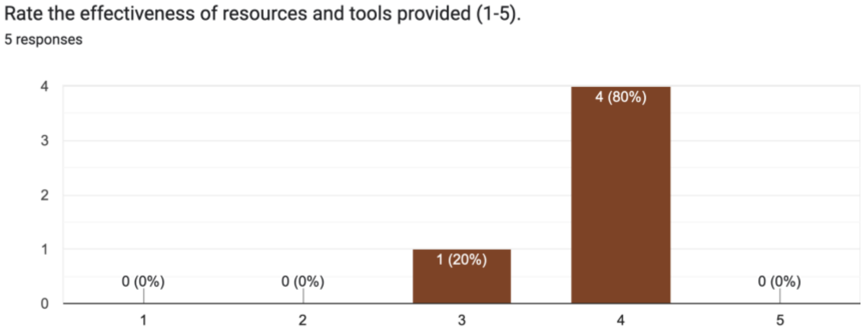


Figure 15: Effectiveness of the resources and tools

- Results
  - Rating 1: 0 responses (0%)
  - Rating 2: 0 responses (0%)
  - Rating 3: 1 response (20%)
  - Rating 4: 4 responses (80%)
  - Rating 5: 0 responses (0%)
- Mean (Average) - 3.8
- Median - 4

Considering that most responses were a rating of 4, with only one deviating from this, the median is the most representative measure for this dataset, effectively communicating the most common respondent experience regarding the effectiveness of resources and tools.

Metric 6: Effectiveness of Project Documentation

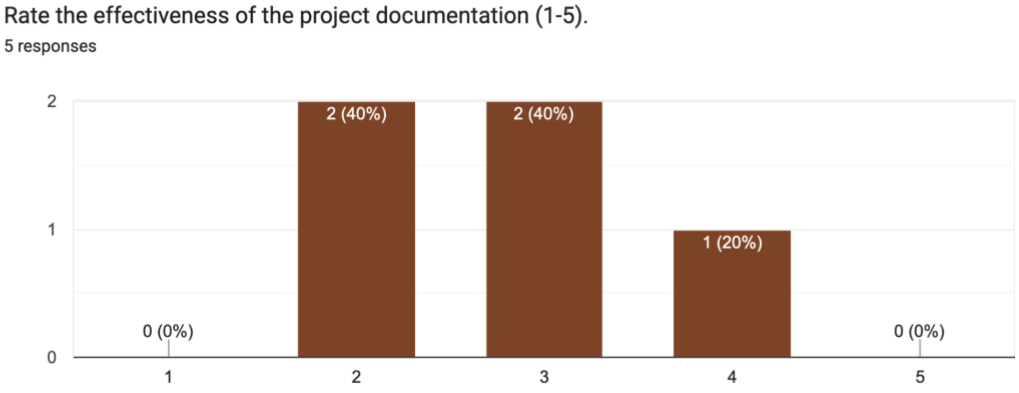


Figure 16: Effectiveness of the project documentation.

- Results
  - Rating 1: 0 responses (0%)
  - Rating 2: 2 responses (40%)

    - Rating 3: 2 responses (40%)
    - Rating 4: 1 response (20%)
    - Rating 5: 0 responses (0%)
- Mean (Average) - 2.8
- Median - 3

The median is the most suitable measure for assessing the effectiveness of project documentation, as it represents the central point of participants' opinions without being skewed by extremes. This approach is particularly relevant for ordinal data spread across the middle of the scale, suggesting that the documentation was found to be moderately effective by the average respondent.

Summary

1. **Time Commitment**: Participants dedicate an average of 3.6 hours per week to the project. The median commitment is 4 hours, suggesting that most team members consistently engage for several hours each week.
2. **Balance and Collaboration**: The median balance between academic and project responsibilities is rated at 2, which may indicate challenges in managing these commitments. Team collaboration is viewed positively, with a median effectiveness rating of 4, reflecting satisfactory teamwork dynamics.
3. **Skill Development**: Technical skill levels show an increase, with the median rating improving from 2 at the start to 3 by the end of the project, indicating progress in participants' technical proficiency.
4. **Resource Effectiveness**: The median response to the effectiveness of resources and tools is 4, demonstrating a favorable reception. Project documentation receives a median effectiveness score of 3, highlighting room for enhancement to better support the team’s needs.

Qualitative Analysis

In the qualitative evaluation of the software development process for LBFE, an analysis was conducted to understand the sentiments, emotions, and perspectives within respondents' verbal feedback. This involved examining frequently mentioned words and phrases within the context of their responses, aiming to capture the nuances of participant experiences, challenges encountered, and overall project engagement. The sentiment analysis sought to uncover prevalent themes and concerns, offering insights into the collective experiences and challenges faced by the LBFE project team through a human-centric approach.

Analysis

Participants' feedback reflected a spectrum of impacts and contributions, underscored by words such as "feel" and "much," indicating varied feelings about their roles and achievements. Notable themes included time management difficulties and the balancing act between project responsibilities and academic demands, highlighted by terms like "time" and "project." The analysis also touched upon the dynamics of team communication and collaboration, acknowledging both its strengths and challenges through words like "team" and "communication." The sufficiency of training and resources, as well as the clarity and utility of project documentation, emerged as areas with room for improvement, evidenced by feedback mentioning "tutorials" and expressing some dissatisfaction. Furthermore, the analysis delved into how team meetings influenced project progression and tackled technical hurdles, with an emphasis on fostering improvement. Words like "backend" and "problem" pointed to specific technical issues faced. Motivations for participating in the project varied among team members, with personal growth and external incentives both playing significant roles. Feedback also suggested avenues for enhancing project structure and workflow, particularly in coding practices and team involvement, and highlighted the importance of aligning personal goals with project objectives, focusing on skill acquisition.

By providing a comprehensive overview that spans collaboration dynamics, communication effectiveness, encountered challenges, and the impact of training and resources, this analysis highlights areas for improvement in the software development process. It emphasizes the importance of empathetic and informed project management strategies to enhance the Agile practices in LBFE, aiming for a more inclusive and effective project experience for all participants.

Summary

Team members in the project experienced a variety of challenges and learning opportunities. Their focus spanned across team communication, technical difficulties, training resources, and project documentation. Notably, there was significant personal growth in technical skills, though balancing the project with academic responsibilities posed challenges. Improvement areas identified include enhancing documentation, refining task management, and streamlining issue resolution processes.

Reflection

My role as an Agile Process Analyst at LBFE has been a journey of learning and professional growth. This experience highlighted the importance of flexibility in software process management, especially when balancing academic and project commitments. I observed firsthand the impact of Agile practices on team dynamics, communication, and overall project progress. My involvement in this role has not only deepened my understanding of Agile methodologies but also reinforced the value of adapting Agile principles to meet diverse project needs.

Conclusion

My tenure across different capacities at HIDE, has been a cornerstone in my professional and personal development journey. Transitioning through roles, from a Technical Consultant to an Agile Process Analyst, I engaged deeply with tasks that not only sharpened my technical acumen but also expanded my understanding of Agile. It has equipped me to adeptly tackle complex technical challenges and underscored the value of teamwork, strategic problem-solving, and applying technical expertise within Agile frameworks to real-world scenarios. As I embark on future endeavors, I am inspired to apply the set of skills, knowledge, and experiences gained from my time at HIDE to innovate user-centric software solutions.

Works Cited

1. Villavicencio, M., Narvaez, E., Izquierdo, E., & Pincay, J. (2017). Learning scrum by doing real-life projects. *2017 IEEE Global Engineering Education Conference (EDUCON)*, 1450-1456. https://doi.org/10.1109/EDUCON.2017.7943039.
2. Azanha, A., Argoud, A., Junior, J., & Antoniolli, P. (2017). Agile project management with Scrum. *International Journal of Managing Projects in Business*, 10, 121-142. https://doi.org/10.1108/IJMPB-06-2016-0054.
3. Hidalgo, E. (2019). Adapting the scrum framework for agile project management in science: case study of a distributed research initiative. *Heliyon*, 5. https://doi.org/10.1016/j.heliyon.2019.e01447.
4. Maxim, B., & Kessentini, M. (2016). An introduction to modern software quality assurance. , 19-46. https://doi.org/10.1016/B978-0-12-802301-3.00002-8.
5. Prior, J., & Leaney, J. (2020). Software Quality and Its Entanglements in Practice. *Ethnographic Praxis in Industry Conference Proceedings*. https://doi.org/10.1111/epic.12024.
6. Ghanbari, H., Vartiainen, T., & Siponen, M. (2018). Omission of Quality Software Development Practices. *ACM Computing Surveys (CSUR)*, 51, 1 - 27. https://doi.org/10.1145/3177746.
7. (2021). A Methodology for Incorporating Quality Assurance Practices during Software Development Life Cycle. *International Journal of Advanced Trends in Computer Science and Engineering*. https://doi.org/10.30534/ijatcse/2021/1141032021.
8. Bhujang, R., & Suma, V. (2018). A comprehensive solution for risk management in software development projects. *Int. J. Intell. Syst. Technol. Appl.*, 17, 153-175. https://doi.org/10.1504/IJISTA.2018.10012891.
9. Kuatov, A., & Hamada, M. (2021). Risk Management in Software Development and Software Engineering Projects. *Computer Science and Information Technology*, 9. https://doi.org/10.36648/2349-3917.21.9.93.
10. Bannerman, P. (2015). A Reassessment of Risk Management in Software Projects. , 1119-1134. https://doi.org/10.1007/978-3-319-05915-0_20.
11. Odzaly, E., Greer, D., & Stewart, D. (2017). Agile risk management using software agents. *Journal of Ambient Intelligence and Humanized Computing*, 9, 823 - 841. https://doi.org/10.1007/s12652-017-0488-2.
12. Kuatov, A., & Hamada, M. (2021). Risk Management in Software Development and Software Engineering Projects. Computer Science and Information Technology, 9. https://doi.org/10.36648/2349-3917.21.9.93.
13. Vakaloudis, A., Cahill, B., O'Leary, C., & Challa, D. (2020). Preparation and execution of final year student projects on the cloud. *2020 IEEE Frontiers in Education Conference (FIE)*, 1-7. https://doi.org/10.1109/FIE44824.2020.9273971.
14. Zhu, W. (2015). Hands-On Network Programming Projects in the Cloud. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/2676723.2677257.
15. Madhav, N., & Joseph, M. (2017). Cloud for Engineering Education: Learning networks for effective student engagement. *2017 IEEE 7th Annual Computing and*

Communication Workshop and Conference (CCWC), 1-4. https://doi.org/10.1109/CCWC.2017.7868344.

16. Al Qurashi, S., & Qureshi, M. R. J. (2014). Scrum of Scrums solution for large size teams using Scrum methodology. *Life Science Journal, 11*(8), 443-449.
17. S., G., & H., J. (2023). Management of Software Projects in Degree Projects using an Agile Methodology. *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6. https://doi.org/10.23919/CISTI58278.2023.10212009.
18. Nagaria, J., Sadath, L., & Ahmed, S. (2019). Agile Implementation-A milestone for Academics using Software Engineering Industry Practices. *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, 1-6. https://doi.org/10.1109/ICASET.2019.8714575.
19. Holvikivi, J., & Hjort, P. (2017). Agile Development in Software Engineering Instruction. , 609-618. https://doi.org/10.1007/978-3-319-74310-3_61.
20. Jez Humble and David Farley. 2010. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (1st. ed.). Addison-Wesley Professional.
21. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for Agile Software Development. Agile Alliance. https://agilemanifesto.org/
22. Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. https://www.scrumguides.org/scrum-guide.html

Title of Appendix

Appendix A: Definitions

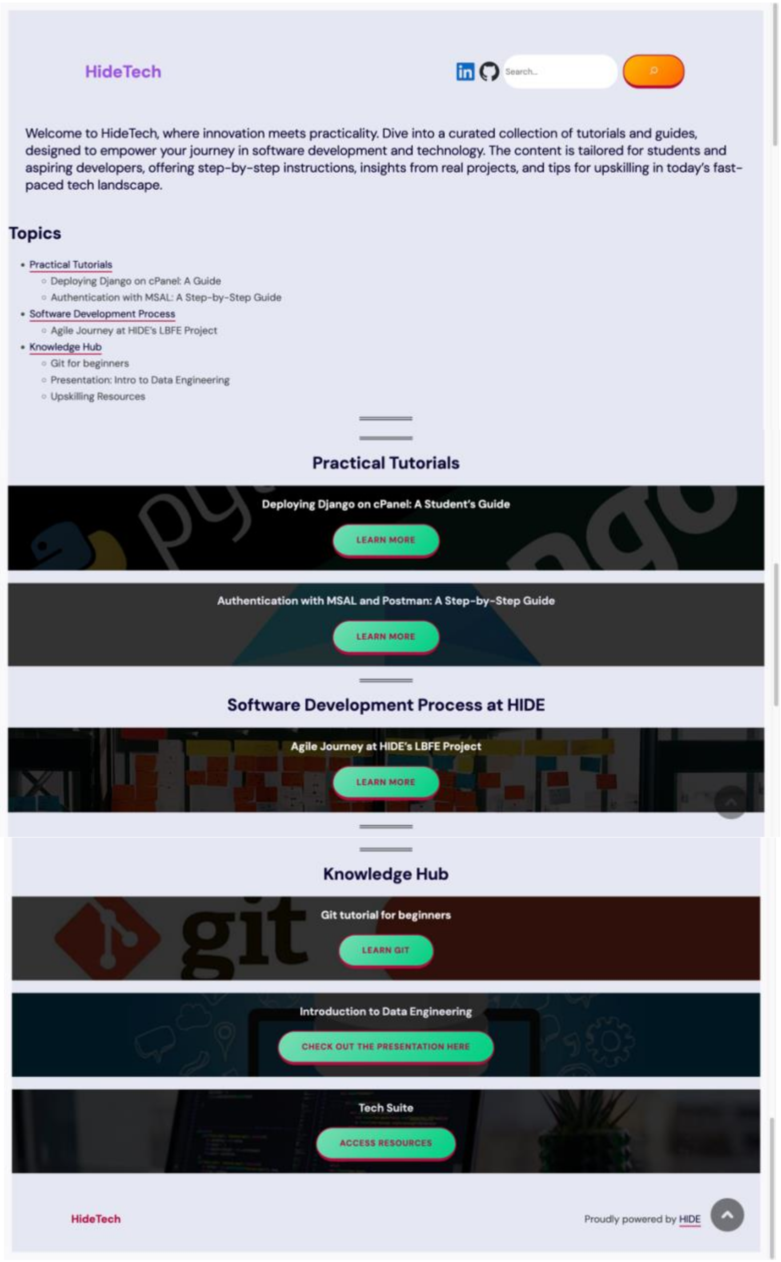
1. **Humane Interface Design Enterprise (HIDE)**: A student-run organization focused on software development and usability research based at Michigan Technological University.
2. **Technical Consultant**: A professional who provides technical guidance and solutions to specific problems, ensuring that projects meet their technological requirements.
3. **Software Development**: The process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components.
4. **Agile Methodology**: A set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers/end users. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages flexible responses to change.
5. **Scrum**: An Agile framework designed to help teams work together, encouraging teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.
6. **IT Operations**: IT Operations emphasizes the automation, monitoring, and maintenance of software in production environments to ensure reliability, efficiency, and faster deployment cycles.
7. **DevOps**: DevOps is a set of practices that combines Software Development (Dev) and IT Operations (Ops) to shorten the development life cycle and deliver high-quality software continuously.
8. **Quality Assurance (QA)**: Quality Assurance (QA) in software engineering is the process of ensuring that software products meet specified standards and requirements through systematic activities and testing.
9. **Django**: A high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development.
10. **cPanel**: A web-based hosting control panel provided by many hosting providers to website owners allowing them to manage their websites from a web-based interface.
11. **Authentication Protocol**: An authentication protocol is a set of rules that determine how two parties prove their identities to each other securely over a network.
12. **Microsoft Authentication Library (MSAL)**: A library for applications to authenticate and obtain tokens from Azure AD (Active Directory) and Microsoft accounts.
13. **Azure Active Directory (Azure AD)**: Microsoft's cloud-based identity and access management service, which helps your employees sign in and access resources.
14. **Vue.js**: An open-source JavaScript framework for building user interfaces and single-page applications.

15. **API (Application Programming Interface)**: A set of rules that allows different software entities to communicate with each other. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, and the conventions to follow.
16. **Postman**: A collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs faster.
17. **OAuth 2.0**: An authorization framework that enables applications to obtain limited access to user accounts on an HTTP service. It allows users to share information about their accounts with third-party applications or websites without having to expose their password.
18. **CI/CD (Continuous Integration/Continuous Deployment)**: A method to frequently deliver apps to customers by introducing automation into the stages of app development, including integration, testing, and deployment.
19. **Jenkins**: An open-source automation server that enables developers to build, test, and deploy their software through continuous integration and facilitates continuous delivery.
20. **GitHub Actions**: A CI/CD feature within GitHub that allows automation of your build, test, and deployment pipeline directly from your GitHub repository.
21. **Cloud Computing**: The delivery of computing services - including servers, storage, databases, networking, software, analytics, and intelligence - over the Internet (“the cloud”) to offer faster innovation, flexible resources, and economies of scale.
22. **AWS (Amazon Web Services)**: A comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure as a service (IaaS), platform as a service (PaaS), and packaged software as a service (SaaS) offering.
23. **Likert Scale**: A Likert scale is a psychometric scale commonly involved in research that employs questionnaires, used to represent people's attitudes or feelings toward a topic with a range of fixed-choice answers on a scale of agreement or frequency.

Appendix B: Hidetech Website

This section of the report features selective screenshots from the ‘hidetech’ website, highlighting key aspects of the project's documentation.

1. **Homepage**



HideTech

Search...

Welcome to HideTech, where innovation meets practicality. Dive into a curated collection of tutorials and guides, designed to empower your journey in software development and technology. The content is tailored for students and aspiring developers, offering step-by-step instructions, insights from real projects, and tips for upskilling in today's fast-paced tech landscape.

Topics

- Practical Tutorials
    - Deploying Django on cPanel: A Guide
    - Authentication with MSAL: A Step-by-Step Guide
- Software Development Process
    - Agile Journey at HIDE's LBFE Project
- Knowledge Hub
    - Git for beginners
    - Presentation: Intro to Data Engineering
    - Upskilling Resources

Practical Tutorials

Deploying Django on cPanel: A Student's Guide

LEARN MORE

Authentication with MSAL and Postman: A Step-by-Step Guide

LEARN MORE

Software Development Process at HIDE

Agile Journey at HIDE's LBFE Project

LEARN MORE

Knowledge Hub

Git tutorial for beginners

LEARN GIT

Introduction to Data Engineering

CHECK OUT THE PRESENTATION HERE

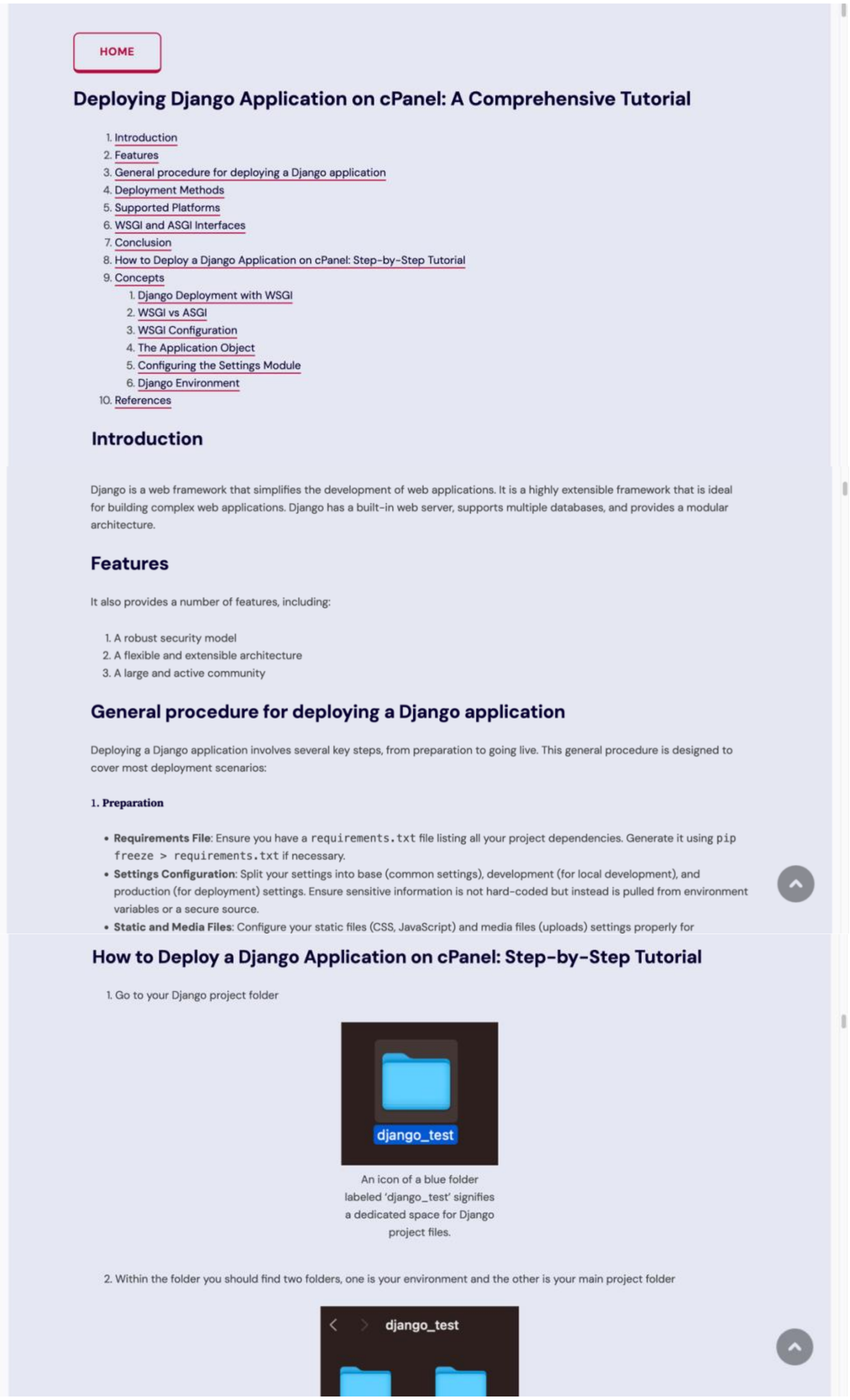
Tech Suite

ACCESS RESOURCES

HideTech

Proudly powered by HIDE

2. Deploying Django on cPanel Page



Deploying Django Application on cPanel: A Comprehensive Tutorial

Introduction

Django is a web framework that simplifies the development of web applications. It is a highly extensible framework that is ideal for building complex web applications. Django has a built-in web server, supports multiple databases, and provides a modular architecture.

Features

It also provides a number of features, including:

1. A robust security model
2. A flexible and extensible architecture
3. A large and active community

General procedure for deploying a Django application

Deploying a Django application involves several key steps, from preparation to going live. This general procedure is designed to cover most deployment scenarios:

1. Preparation

- **Requirements File**: Ensure you have a `requirements.txt` file listing all your project dependencies. Generate it using `pip freeze > requirements.txt` if necessary.
- **Settings Configuration**: Split your settings into base (common settings), development (for local development), and production (for deployment) settings. Ensure sensitive information is not hard-coded but instead is pulled from environment variables or a secure source.
- **Static and Media Files**: Configure your static files (CSS, JavaScript) and media files (uploads) settings properly for

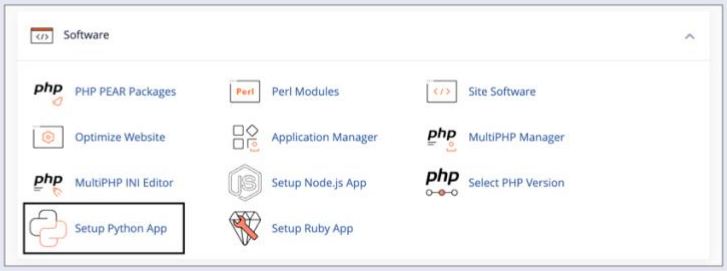
How to Deploy a Django Application on cPanel: Step-by-Step Tutorial

1. Go to your Django project folder

An icon of a blue folder labeled 'django_test' signifies a dedicated space for Django project files.

2. Within the folder you should find two folders, one is your environment and the other is your main project folder

9. Navigate to cPanel home page and scroll down to the "**Software**" section. Click on "**Setup Python App**"
    - If the "**Setup Python App**" feature is not visible within your cPanel interface, I recommend reaching out to the IT department for assistance in configuring your cPanel to enable this feature.
        - IT: **it-help@mtu.edu**



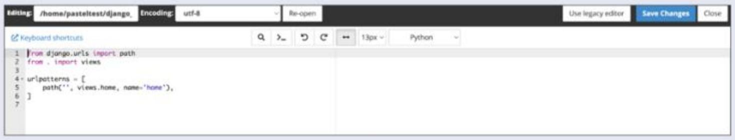
The 'Software' section of a web hosting control panel interface, highlighting various tools available for website optimization and application setup, including 'Setup Python App' highlighted for setting up a Python application on the server.

10. Click on "**Create Application**" button. As you can see I have already created an application.

Django Environment Detection

This Python snippet below detects whether the Django web application is running in a production or development environment. It relies on the Django settings module to determine the environment based on the value of the DEBUG setting.

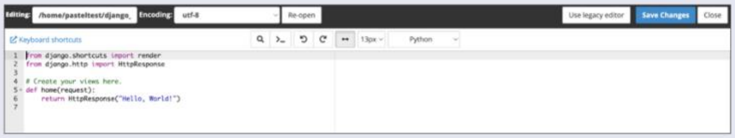
```
from django.conf import settings

def is_production():
    """
    Checks if the environment is production.
    Returns:
        True if the environment is production, False otherwise.
    """
    return settings.DEBUG is False

def is_development():
    """
    Checks if the environment is development.
    Returns:
        True if the environment is development, False otherwise.
    """
    return settings.DEBUG is True

if is_production():
    what_is_the_env = 'DEV'
    print(what_is_the_env)
else:
    what_is_the_env = 'PROD'
    print(what_is_the_env)
```



```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
]
```

A screenshot showing a Python code editor with a file being edited. The file contains import statements and a URL pattern list for a Django web application. The code imports the 'path' function from 'django.urls' and 'views' from the current directory. It defines a single URL pattern that maps the root URL (") to a view function named 'home', with the name 'home' for reverse URL matching.

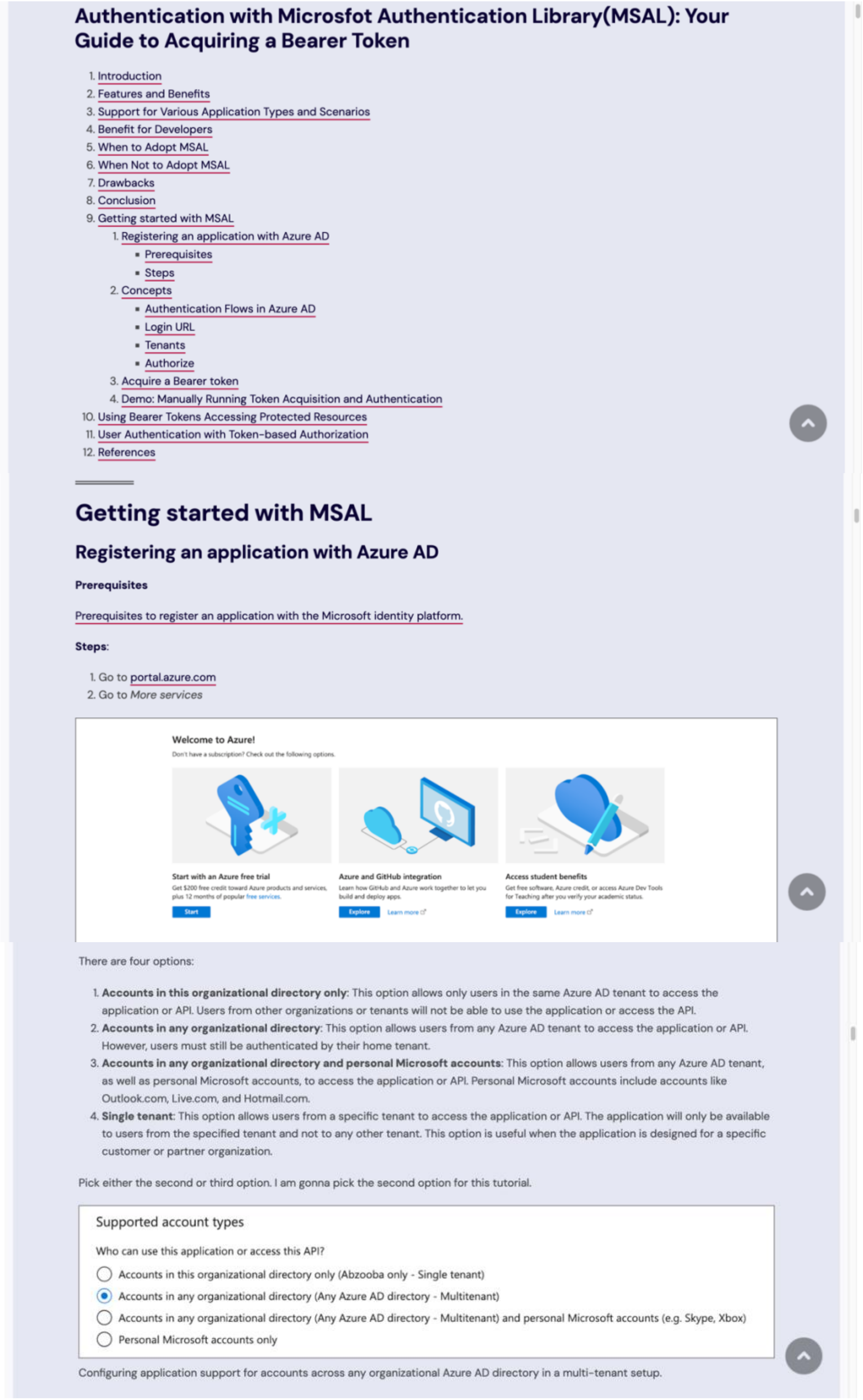
28. In the '*views.py*' file you will find all the views and their associated methods of your application.



```
from django.shortcuts import render
from django.http import HttpResponse

# Create your views here.
def home(request):
    return HttpResponse("Hello, World!")
```

A screenshot of a Python code editor showing the contents of a views.py file in a Django application. The file contains import statements for Django's 'render' shortcut and 'HttpResponse' class. A function named 'home' is defined, which takes a 'request' parameter and returns an 'HttpResponse' object with the text 'Hello, World!'. This function is intended to be a view that outputs a simple greeting when a user visits the home page of the web application.

3. Authentication with MSAL and Postman Page



Authentication with Microsfot Authentication Library(MSAL): Your Guide to Acquiring a Bearer Token

Getting started with MSAL

Registering an application with Azure AD

Prerequisites

Prerequisites to register an application with the Microsoft identity platform.

Steps:

1. Go to portal.azure.com
2. Go to *More services*

Welcome to Azure!

Don't have a subscription? Check out the following options.

Start with an Azure free trial

Get $200 free credit toward Azure products and services, plus 12 months of popular free services.

Azure and GitHub integration

Learn how GitHub and Azure work together to let you build and deploy apps.

Access student benefits

Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.

There are four options:

1. **Accounts in this organizational directory only**: This option allows only users in the same Azure AD tenant to access the application or API. Users from other organizations or tenants will not be able to use the application or access the API.
2. **Accounts in any organizational directory**: This option allows users from any Azure AD tenant to access the application or API. However, users must still be authenticated by their home tenant.
3. **Accounts in any organizational directory and personal Microsoft accounts**: This option allows users from any Azure AD tenant, as well as personal Microsoft accounts, to access the application or API. Personal Microsoft accounts include accounts like Outlook.com, Live.com, and Hotmail.com.
4. **Single tenant**: This option allows users from a specific tenant to access the application or API. The application will only be available to users from the specified tenant and not to any other tenant. This option is useful when the application is designed for a specific customer or partner organization.

Pick either the second or third option. I am gonna pick the second option for this tutorial.

Supported account types

Who can use this application or access this API?

○ Accounts in this organizational directory only (Abzooba only - Single tenant)
● Accounts in any organizational directory (Any Azure AD directory - Multitenant)
○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
○ Personal Microsoft accounts only

Configuring application support for accounts across any organizational Azure AD directory in a multi-tenant setup.

Let's understand few concepts and methodologies first before we proceed forward.

Authentication Flows in Azure Active Directory (Azure AD)

Authentication flows in Azure AD refer to the different methods that can be used to authenticate a user and obtain an access token for accessing protected resources.

1. **Authorization Code Flow**

The authorization code flow is used for web applications and involves the user being redirected to the Azure AD login page, where they enter their credentials. After successful authentication, Azure AD returns an authorization code to the application, which can then be exchanged for an access token.

2. **Implicit Grant Flow**

The implicit grant flow is also used for web applications but is intended for public clients, such as single-page applications. In this flow, the access token is returned directly to the application after the user successfully authenticates with Azure AD.

3. **Device Code Flow**

The device code flow is used for devices that cannot display a traditional login page, such as game consoles or smart TVs. Users are provided with a code to enter on a separate device, such as a mobile phone or computer, which allows them to authenticate and obtain an access token.

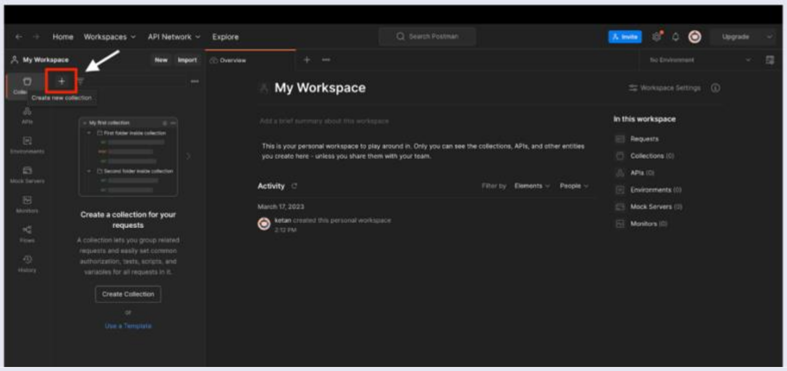
4. **Resource Owner Password Credentials (ROPC) Flow**

Demo: Manually Running Token Acquisition and Authentication

Here's a demo of manually acquiring the token using Postman. At this point you should have Postman installed.

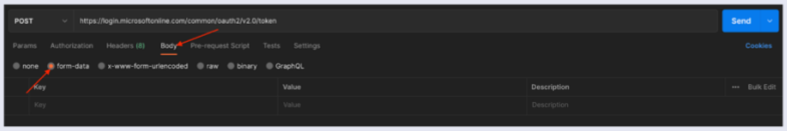
***GET* Request**

1. Open Postman. Go to Workspaces > My Workspace. Click New > New Collection



This is your personal workspace to play around in. Only you can see the collections, APIs, and other entities you create here - unless you share them with your team.

13. We will send query params as form data. Go to Body, check *'form-data'*

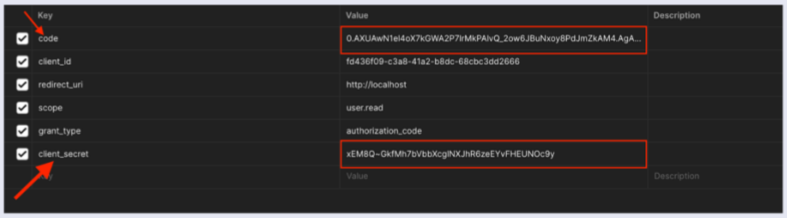


POST https://login.microsoftonline.com/common/oauth2/v2.0/token

Body: ○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

Postman request setup for OAuth 2.0 token retrieval from Microsoft identity platform.

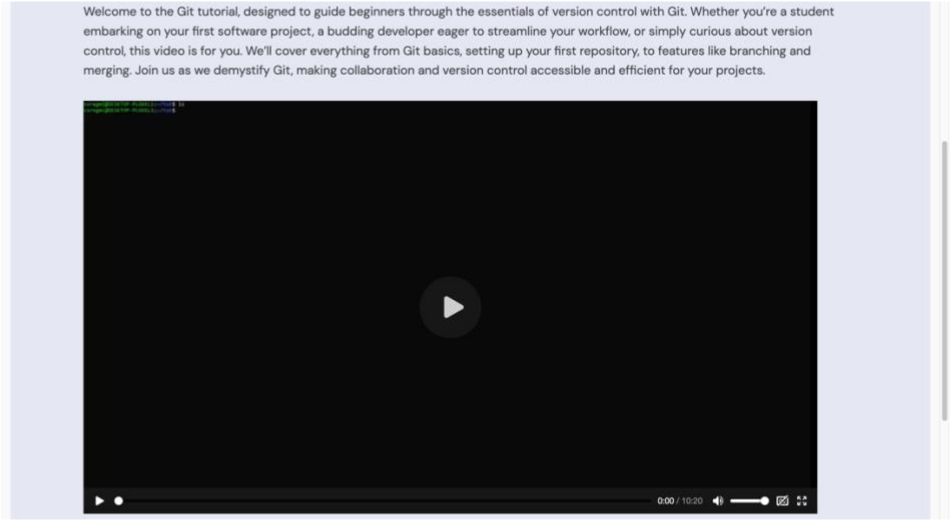
14. Start filling in the keys and values. First, paste the *code* we copied in the parameters. Please note that this time we have a new parameter in our list called *client_secret* as you can see in the image below. If you remember we have saved the *client_secret*. Copy and paste it next to *client_secret* parameter.



	Key	Value	Description
☑	code	0.AXUAwN1el4oX7kGWA2P7lrMkPAlvQ_2ow6JBuNxoy8PdJmZkAM4.AgA...	
☑	client_id	fd436f09-c3a8-41a2-b8dc-68cbc3dd2666	
☑	redirect_uri	http://localhost	
☑	scope	user.read	
☑	grant_type	authorization_code	
☑	client_secret	xEM8Q~GkfMh7bVbbXcgINXJhR6zeEYvFHEUNOc9y	
	Key	Value	Description

Configuring authorization details in Postman for MSAL token acquisition.

4. **Git Tutorial Page**



Welcome to the Git tutorial, designed to guide beginners through the essentials of version control with Git. Whether you're a student embarking on your first software project, a budding developer eager to streamline your workflow, or simply curious about version control, this video is for you. We'll cover everything from Git basics, setting up your first repository, to features like branching and merging. Join us as we demystify Git, making collaboration and version control accessible and efficient for your projects.

Appendix C: Software Development Process Survey

The following section includes a compilation of both qualitative and quantitative questions used in the Google survey conducted with the LBFE team.

Quantitative Evaluation Questions

1. Approximately how many hours per week do you dedicate to the LBFE project, including meetings and other project-related activities?
2. Rate the balance between your academic responsibilities and project work (1-5), considering how one affects the other.
3. Rate the team collaboration and dynamics (1-5)
4. Approximately how many defects or bugs have you encountered throughout your time at LBFE project this semester?
5. On average, how quickly are bugs/issues resolved and suggestion on how can bug/issue resolution be effectively implemented?
6. How many tasks are currently in your backlog?
7. Rate the fairness of task allocation and distribution in the team (1-5)
8. How many total tasks were assigned to you on Trello and approximately how many of them have you completed during this project?
9. Rate your technical skill level at the start of the project (1-5)
10. Rate your technical skill level at the end of the project (1-5)
11. Rate the effectiveness of resources and tools provided (1-5)
12. Rate the effectiveness of the project documentation (1-5)

Qualitative Evaluation Questions

1. Please elaborate on your overall experience with this project. How do you feel your contributions have impacted its outcomes?

2. Describe the key challenges you've faced while balancing project responsibilities with your academic commitments. How have you managed these challenges?
3. Can you provide details on how effective you find team communication and collaboration within the project? Include any specific instances that highlight these aspects.
4. Discuss the helpfulness of the training and resources provided for this project. Were there any areas where you felt additional support was needed?
5. Describe the clarity and usefulness of the project documentation. Were your tasks and roles clearly defined and helpful?
6. Reflect on how team meetings have facilitated project progress. In what areas do you feel you could improve your involvement and how can others do the same?
7. Describe the most significant technical challenges you encountered during the project. How did you and your team address these challenges?
8. What motivates you to participate in the LBFE project? Describe any additional support that could enhance your commitment and productivity.
9. What specific changes to the project structure or workflow would you suggest encouraging more effective development and participation?
10. What personal goals do you hope to achieve through your participation in the LBFE project? How does this project align with your goals and aspirations?