Data migration of databases and the implementation of Django, a web framework software

Olivia Arredondo, Hartnell Community College

Mentors: Mike McCann, Jenny Paduan, Joe Gomez

Summer 2015

Keywords: Django, Database, Data migration, Python

ABSTRACT

For the Monterey Bay Aquarium Research Institute (MBARI), thousands of animal and rock samples are collected every year using deep sea vehicles. The specimens and associated data retrieved on the cruise are recorded in the Expedition and Samples Databases. Web interfaces to the databases were developed using Microsoft's Active Server Pages (ASP) on the server Typhoon. A new web server named Wind is scheduled to replace the previous one but failed to support the outdated applications needed by ASP as initially configured. The project's goal was to debug the Samples and Expedition Database web pages while testing another software that could potentially replace ASP. To debug the ASP files required testing all of the Samples Database links and if a problem occurred, go to the code to find the error. For the second part of the project, a replacement for ASP called Django (a free, open source web framework) was found and tested. To test whether Django would be a good fit for the databases, time had to be taken in order to learn the software and download all the supporting applications needed. Afterwards, I had to determine whether Django

can work for the databases. Django is a superior option in comparison to ASP because it requires far less code, focuses on Python rather than multiple programming languages, and many resources are available online. By improving the user interface, researchers who utilize the databases can efficiently gather and record data for their work.

INTRODUCTION

The Monterey Bay Aquarium Research Institute (MBARI) is dedicated to the research of oceans around the world. In order to advance in research, samples and specimens must be taken from the ocean for analysis. Some samples collections, both geological and biological, are recorded onto video on board the ship and later transferred to the Samples Database. All cruises have their data, such as video annotations and frame grabs, water column physical data, and vehicle navigation, as well as links to the samples, stored in the Expedition Database.

Both the Samples and Expedition Databases were created with the intention of allowing MBARI researchers an accessible way to document cruises and their findings. In the Samples Database, tens of thousands of sample records are stored in the database with their sample ID, dive number, way point name, chief scientist, collector, longitude, latitude, depth, equipment, and analysis type. The user can look for specific samples by filling out the web query with specific details they are looking for. After initializing the search, a result page will appear with samples that meet the query. The Expedition Database was made with the purpose of planning and searching expeditions. The database contains past, present, and future expeditions along with details about the ship used, comments recorded, equipment needed, people on the ship, date, dive number, accomplishments, purpose, and site.

The databases were created in the late 1990's by MBARI staff. The user interface of the websites were programmed utilizing Active Server Pages (ASP)

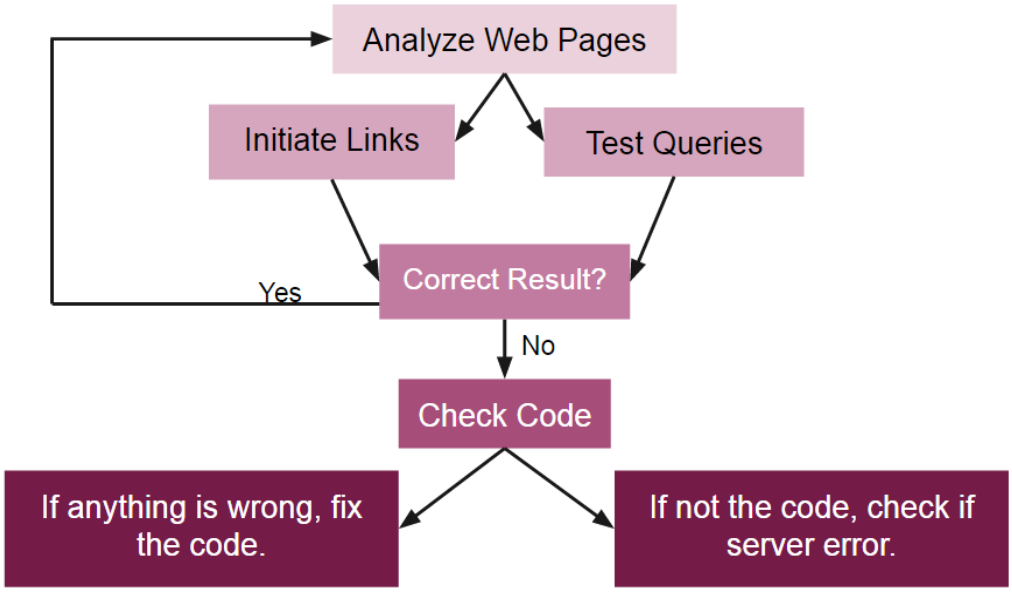
software. ASP is a web framework software created by Microsoft during the late 1990's (), which allows coding with different scripting languages to construct queries to the databases and hypertext markup language (HTML) to format the results of the queries on web pages.

MATERIALS AND METHODS

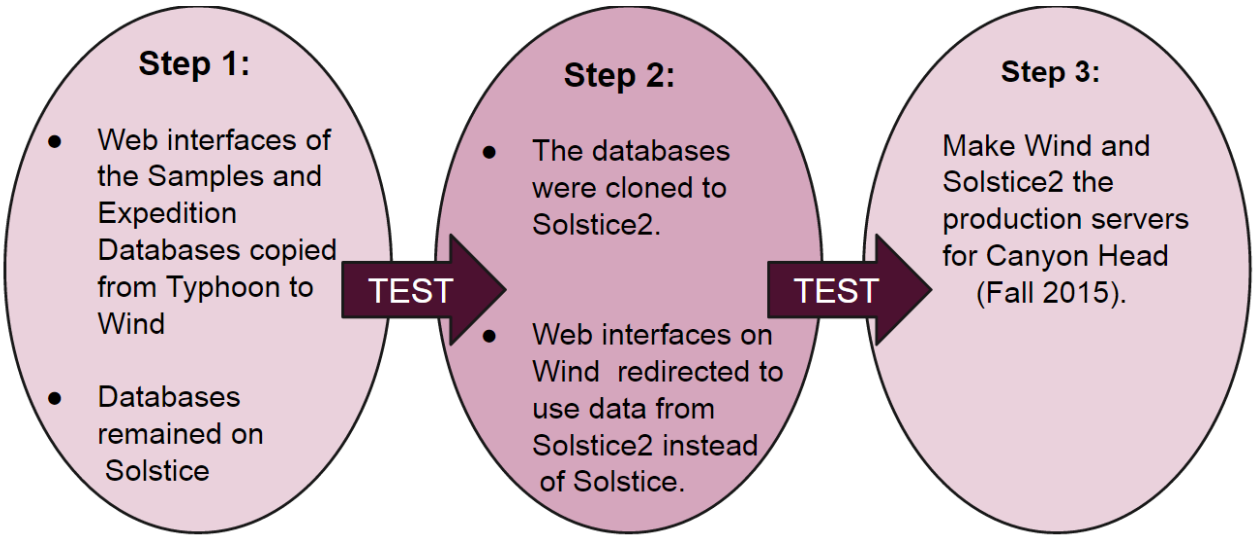
PART 1: DEBUGGING THE SAMPLES AND EXPEDITION DATABASE

The materials utilized were Dreamweaver, Remote Desktop, and ASP. In order to do this part I had to utilize a web page code editor called Dreamweaver (Adobe Systems, Inc.) to allow me to view and edit the code. The program also allowed me to have a visual of what the website would appear to the user. I used Remote Desktop (Microsoft), which acts as a virtual machine and helps to manage servers, to view more details for the error pages that show up from certain links and queries.

First, on the web pages for the databases, all of the links and queries were tested. If the links/ queries came up with an error message or led to the wrong page, the correlating ASP file was analyzed using Dreamweaver. If there was any error in the code or supporting files, then it was edited. If not, then the IS department was contacted to determine if the issue lay with settings on the server.



The flowchart process of debugging begins with finding web page errors and fixing either the code or server settings.



The step by step process of migration is done in order to prevent confusion of which migration causes which bug. In between each step is testing that is displayed in the previous diagram.

PART 2: IMPLEMENTING DJANGO

First order of business was to download a virtual machine program called "VirtualBox" to act as an environment in which I could use Django. Time also had to be taken in order to download Django as well as the supporting applications needed for Django to run efficiently. When that was completed GitHub had to be connected to Django through the STOQS (Spatial Temporal Oceanographic Query System) Database project so whatever is produced on the command line goes to the GitHub account.

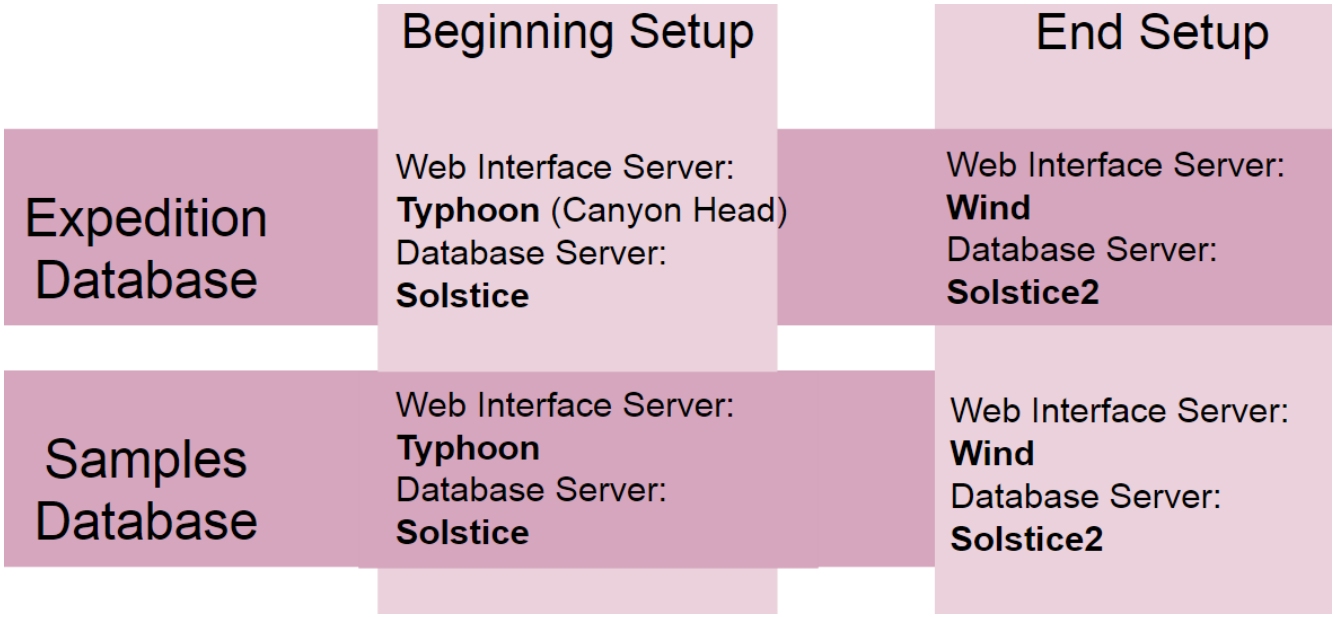
RESULTS

PART 1: DEBUGGING THE SAMPLES AND EXPEDITION DATABASE

Most, if not all, of the notable bugs have been noted. Most have been fixed besides an issue with the image selection page on the Samples Database. The issue occurs whenever a sample is chosen to be viewed on the "Sample Selection" page. On the page is a link that reads "View All Sample Images". The link leads to a page that displays every picture of that particular sample. However, if there are no pictures available to view, an error page that was programed to say "No images available" is supposed to come up. In actuality, a 500 error page appears instead.

Besides the sample image issue, the web pages are fully functional. The user, MBARI employees and collaborators, should not be greatly impacted by the image bug. The user can easily use the seach page for both the Samples Database and Expedition

Database without complications. The links available do in fact lead to the correct



destination.

The web pages for the Samples and Expedition Database are composed of two servers. One server is for the web interface and the other for databases. The original web interface server was Typhoon which is to replaced by Wind. The original database server Solstice is to replaced by Solstice2.

PART 2: IMPLEMENTING DJANGO

First order of business was to download a virtual machine program called "VirtualBox" to act as an environment I could use Django in. Time also had to be taken in order to download DJango as well as the supporting applications needed for Django to run efficiently. When that was completed GitHub had to be connected to Django through the STOQS (Spatial Temporal Oceanographic Query System) Database project so whatever is produced on the command line goes to the GitHub account.

Figures should be inserted into this word document

Figure legends (should be complete sentences and explain the figure completely) (Normal, Times New Roman, 10 pt)

DISCUSSION

PART 1: DEBUGGING THE SAMPLES AND EXPEDITION DATABASE

The Expedition Database had far fewer problems than the Samples Database. The only issue with the Expedition Database was broken links which needed to be fixed through editing the imported files controlling the link's address path. This database is fully ready to be put on the production server to be used by MBARI employees.

The Samples Database also had broken links but many remaining problems are due to outdated programs the links such as Google Earth and 3D Replay. Considering the age of 3D Replay and the current use of STOQS, it can be removed and replaced. The Samples Database image selection page has the most bugs that are code related. An unresolved one is that the image page leads to an error whenever the user selects a sample in which there are no images present.

PART 2: IMPLEMENTING DJANGO

Django took approximately 3 weeks to learn and setup. For someone who lacks the background knowledge of basic web design and object oriented programming, this amount of time would be of no surprise. While Django is a fast, efficient way to set up a web page, experience is needed to use it to the fullest extent. Another concern is is its reliance on third party applications, which are heavily influenced by whether the small group (or individual) consistently debugs and updates said program. However, because GitHub is heavily used with Django, programs can be fixed by anyone from around the world (with the creator having the final say of course). For the databases ASP files, they were a

combination of Perl, VB Script, Java Script, and HTML. This patchwork-style coding method can make data migration highly difficult in comparison to Django's one use of Python only.

CONCLUSIONS / RECOMMENDATIONS

The Expedition Database had far fewer problems than the Samples Database. The only issue that lied with the Expedition Database were broken links which needed to be fixed through editing the imported files controlling the link's pathway. This database is fully ready to be put on the production server to be used by MBARI employees. The Samples Database will still require some debugging in regard to the image issue. If any other issue arises from the web pages in the future then action must be taken to fix the code or server.

Django took approximately 3 weeks to learn and setup. For someone who lacks the background knowledge of basic web design and object oriented programming, this amount of time would be of no surprise. While Django is a fast, efficient way to set up a web page, experience is needed to use it to the fullest extent. Another concern is is its reliance on third party applications, which are heavily influenced by whether the small group (or individual) consistently debugs and updates said program. However, because GitHub is heavily used with Django, programs can be fixed by anyone from around the world (with the creator having the final say of course). For the databases ASP files, they were a combination of Perl, VB Script, Java Script, and HTML. This Frankenstein-like coding method can make data migration highly difficult in comparison to Django's one use of Python only.

The Samples Database also had broken links but many problems were due to outdated programs the links such as 3D Replay. Considering the age of 3D Replay and the current use of STOQS, it can be removed and replaced. If the web

pages are rewritten onto Django, then the use of Google Earth and 3D Replay are not needed. The geological visualization STOQS uses can easily be incorporated into the web pages without the use being removed from the page. This way, the user could see all the information as well as the geology of the sample's/ cruise's location.

ACKNOWLEDGEMENTS

I would like to thank all my mentors for their help and support. All played an important part in my project. Jenny Paduan explained the background of the data stored in the databases. Joe Gomez assisted with any server errors that showed up during the database debugging period. Mike McCann was more than helpful during the set up for the virtual machine and Django.

The greatest thanks goes to my family for their complete support, without their encouragement I wouldn't be here. I also like to thanks Jose Ramirez for always being there for me when I needed him the most. Without

References:

Anon, The Official Microsoft ASP.NET Site. *The Official Microsoft ASP.NET Site*. Available from: http://www.asp.net/.