9/23/2016

REPORT FOR INTERNSHIP AT PIONEER

DHEERAJ P SRIVATSAV
CPM_SC 8085: PROBLEMS IN COMPUTER SCIENCE
SUMMER 2016

Software Developer Internship Report

DuPont Pioneer(SUMMER 2016)

Dheeraj P. Srivatsav
University of Missouri
Department of Computer Science
Columbia, MO
dpstm3@mail.missouri.edu

***Abstract*— My three-month Software Developer Internship was with the Application Development team in DuPont Pioneer at Johnston, IA. I was involved with the precision software the company develops to help the agricultural sector called Encirca, all of my work will be outlined in this report. Experienced and embraced the AGILE method of Software Development Life Cycle. It was a very good learning experience and also understanding the nuances of software development process will prove helpful in my future career path.**

I. Introduction

This report will cover some background information on the project I was involved in, the whole development process which was followed as well as details on how I went about solving the tasks assigned to me. There were three important projects I worked on, first was a code refactoring project helping to reduce the codebase size improve code readability and maintenance. Next, I worked on sprint based stories/tasks with the team to add new features, modify features to the Encirca product and also fix bugs as suggested by business requirements in a timely manner.

II. OVERVIEW

A. *Encirca background information*

DuPont Pioneer is a leading agricultural/chemical conglomerate in North America. They deal with Corn, Cotton and Soybean varieties. They are a major producer of genetically modified organisms (GMOs), including genetically modified crops with insect and herbicide resistance. They develop custom precision software to help the agricultural sector using GIS mapping. The specific software the team focused was Encirca. Encirca combines the latest technology for weather, soils, agronomy and analytics to help maximize crop yields and reduce risks. The software helps farmers make decisions, the more they know about their soils, the better they can manage their inputs. Through a collaboration with University of Missouri and the U.S. Department of Agriculture, Encirca services gives them an advanced 3-D view into the composition of the soils on the farm, including valuable information on soil depth, texture, organic matter content and water holding capacity. Encirca services high resolution soil maps and prediction which have been shown to provide a better basis for decision-making than normal soil maps.

B. *Team Setting and Dynamics*

The Encirca software was worked on by 5 teams, I was part on the team called Piranha. We catered to the specific service area such as Yield and Fertility. The team comprised of 9 developers, 4 Quality Analyst's, 1 Requirement Analyst and a Scrum Master. An intern was added to each of the teams for the summer, the main intention to hire interns is to provide them with valuable industry experience and test their problem solving skills. The intern is paired with a mentor so that his progress and work can be monitored. They follow AGILE/Scrum methodology of Software Development where teams worked on scheduled amount of stories (features) for two weeks(called sprints) and then they would release the code to production.Continuous delivery and maintenance helps keep developers challenged and the product up to date. The code base was MVC architecture and had a C#/.NET & MS SQL platform supporting AngularJS/HTML front end framework.

III. PROJECT -I Fixing Bugs From Backlog

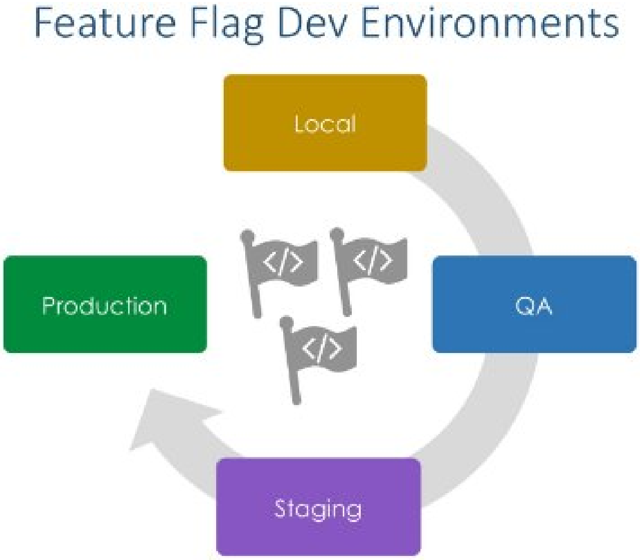
After setting up the development environment and getting to know the product and the features of the product the team will mainly work on, I was assigned to pick up bugs logged in the backlog and fix them. This was mainly to test my problem solving skills and also to get involved with the code base and understand the semantics. This part of the project involved gaining a good understanding of the Encirca product. Some of them were UI-design bugs and others involved data manipulation and logic changes. My task was to

- Understand and recreate the bug
- Implement a non-breaking solution
- Get the code-reviewed and push up the change to Git repository
- Document the cause and solution for future reference.

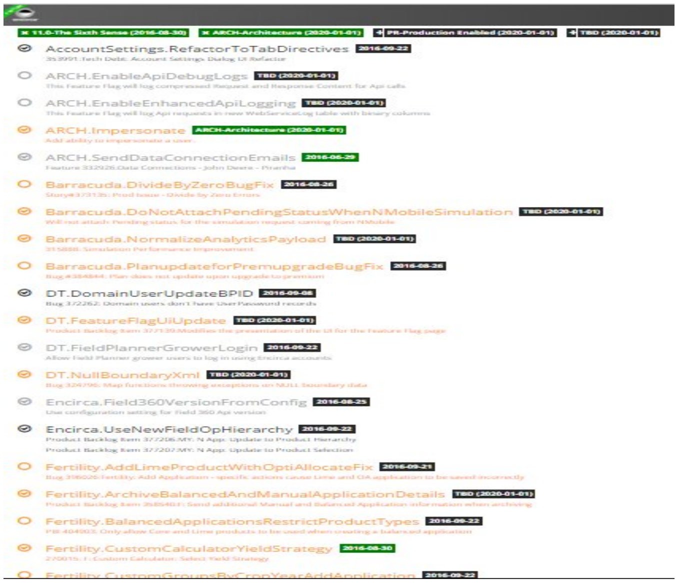
Working on this project helped me understand tools like *Jenkins*, which helped code move from local machine to build into development and test environment. *Crucible* was used for submitting code reviews, which helped getting inputs from the mentor as well as tech lead and make changes from suggestions, if any. During this process I worked on around 10-15 bugs ranging from low to high priority, thereby helping the team clear the bug backlog by at least 30%.

IV. Feature Flag Driven Development

While working in a continuous development and integration environment, it is important to think ahead to avoid running into problems. The process of feature flagging involves you to wrap your features in conditionals that determine who can see your features and when. This is essential if we want to support a Continuous Deployment strategy and need to move unreleased features, or un-tested code into a production environment, but want to turn it off.



As a developer we are required to make sure our code is safely wrapped behind a feature flag and it can be brought on / released by flagging the feature usinf a toggle. This makes developers to be more responsible with the coding habits and develop non-breakable changes. At Encirca we used to toggle the features as shown below, the colored features are the ones which were turned “ON” while the rest were in the “OFF” state.



✖ 11.0-The Sixth Sense (2016-08-30) ✖ ARCH-Architecture (2020-01-01) + PR-Production Enabled (2020-01-01) + TBD (2020-01-01)

AccountSettings.RefactorToTabDirectives 2016-09-22
353991:Tech Debt: Account Settings Dialog UI Refactor

ARCH.EnableApiDebugLogs TBD (2020-01-01)
This Feature Flag will log compressed Request and Response Content for Api calls

ARCH.EnableEnhancedApiLogging TBD (2020-01-01)
This Feature Flag will log Api requests in new WebServiceLog table with binary columns

ARCH.Impersonate ARCH-Architecture (2020-01-01)
Add ability to impersonate a user

ARCH.SendDataConnectionEmails 2016-06-29
Feature 332926:Data Connections - John Deere - Piranha

Barracuda.DivideByZeroBugFix 2016-08-26
Story#373135: Prod Issue - Divide by Zero Errors

Barracuda.DoNotAttachPendingStatusWhenNMobileSimulation TBD (2020-01-01)
Will not attach Pending status for the simulation request coming from NMobile

Barracuda.NormalizeAnalyticsPayload TBD (2020-01-01)
315888: Simulation Performance Improvement

Barracuda.PlanupdateforPremupgradeBugFix 2016-08-26
Bug #384844: Plan does not update upon upgrade to premium

DT.DomainUserUpdateBPID 2016-09-08
Bug 372262: Domain users don't have User Password records

DT.FeatureFlagUiUpdate TBD (2020-01-01)
Product Backlog Item 377139:Modifies the presentation of the UI for the Feature Flag page

DT.FieldPlannerGrowerLogin 2016-09-22
Allow Field Planner grower users to log in using Encirca accounts

DT.NullBoundaryXml TBD (2020-01-01)
Bug 324796: Map functions throwing exceptions on NULL boundary data

Encirca.Field360VersionFromConfig 2016-08-25
Use configuration setting for Field 360 Api version

Encirca.UseNewFieldOpHierarchy 2016-09-22
Product Backlog Item 377206:MY: N App: Update to Product Hierarchy
Product Backlog Item 377207:MY: N App: Update to Product Selection

Fertility.AddLimeProductWithOptiAllocateFix 2016-09-21
Bug 396026:Fertility: Add Application - specific actions cause Lime and OA application to be saved incorrectly

Fertility.ArchiveBalancedAndManualApplicationDetails TBD (2020-01-01)
Product Backlog Item 358540 : Send additional Manual and Balanced Application information when archiving

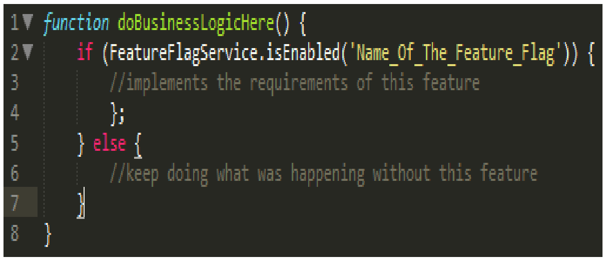
Fertility.BalancedApplicationsRestrictProductTypes 2016-09-22
PBI 404903: Only allow Core and Lime products to be used when creating a balanced application

Fertility.CustomCalculatorYieldStrategy 2016-08-30
270015: F: Custom Calculator: Select Yield Strategy

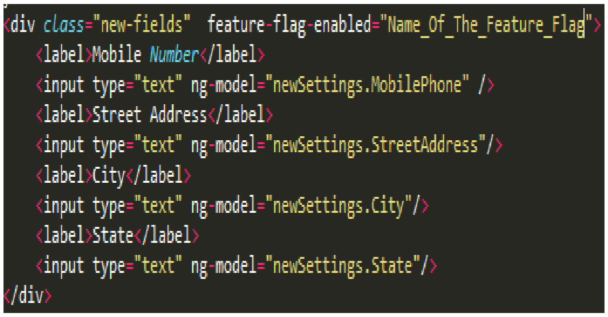
Fertility.CustomGroupsByCropYearAddApplication 2016-09-22

A. *Syntax for Feature Flagging.*

The snippets show the example syntax used for writing code using Feature Flags in JavaScript and HTML files. This was done even for unit tests and back-end C# code.



```
function doBusinessLogicHere() {
    if (FeatureFlagService.isEnabled('Name_Of_The_Feature_Flag')) {
        //implements the requirements of this feature
        };
    } else {
        //keep doing what was happening without this feature
    }
}
```



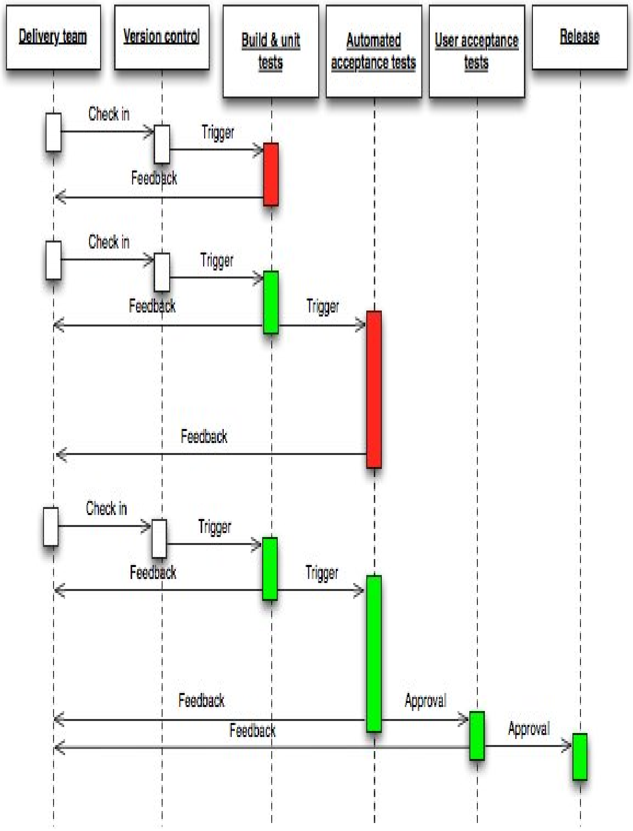
```
<div class="new-fields"  feature-flag-enabled="Name_Of_The_Feature_Flag">
    <label>Mobile Number</label>
    <input type="text" ng-model="newSettings.MobilePhone" />
    <label>Street Address</label>
    <input type="text" ng-model="newSettings.StreetAddress"/>
    <label>City</label>
    <input type="text" ng-model="newSettings.City"/>
    <label>State</label>
    <input type="text" ng-model="newSettings.State"/>
</div>
```

The original approach of this kind of development process was made popular by Flickr. This gives a means to disable features when code is broken in production website, to reduce the query load on the database or on other parts of the system or release certain features to select customers at select times.

B. *Continuous Integration and Deployment*

Continuous integration is all about the entire team committing to master branch daily in the GitHub. A developer will write his custom unit tests to make sure his code is bug free for that particular unit being implemented and then checks-in the code to the master from his local branch. The application should always build and pass all the unit tests and include the check-ins made by all the sub-teams and developers simultaneously. Continuous Deployment is our application should pass the build and integration tests and be deployed to a testing environment and later subsequently should be released to productions by using an automated process.

This is a very important feature in the modern software development life cycle. So there was huge responsibility as an Intern to make sure I was implementing modules which were safe and not breaking any builds. For example, Facebook and LinkedIn go to production at midnight every day, at Encirca we released to production at the end of every sprint, unlike every 6 weeks as it was done before I joined. So, the amount of automation required is massive to go into production that frequently. On a positive note this keeps the developers active and makes it easy to detect and fix bugs in the software. The figure gives an overview of the process.



V. PROJECT II- CODE REFACTORING

The next big task I was asked to implement was to refactor an existing code base which was not modified for a considerable amount of time and its current implement was considered a tech debt. The team wanted me to try to refactor it make sure it was easy to maintain thereafter so that a planned set of new features could be added to that module in the future. I was explained about what the problem was and what the future scope of that project would be. I was asked to come up with a plan and present it to the mentor. I personally liked this feedback process because had they asked me to implement directly, I would have chosen a wrong route on implementing this and wasted some time.

1) *The problem description:*

The Account Settings modal in the application was one of the first written modules and was growing into a big module when new features were added recently. The team decided that it was poorly maintained and adding new features in the future would be difficult, so they wanted to refactor the module in a way which would help it become more robust.

2) *Original Work flow:*

I went through the code and came up with a plan of rewriting the core logic only in different controllers and directives and keeping the same view model. I had to present this to my mentor, he went through the proposed solution and said it might not be a very robust solution and we can still make it better and we brainstormed the whole problem on a white board along with the original code base up.

3) *New Workflow:*

In the new workflow we decided to implement the concept of SOLID principles. Each class or module will cater to a particular business logic or business functionality only. This was one of the main concepts of OOP technologies. This is where I learnt a lot about how to deal with Dependency Injections and Factory Services. I decided to split each tab on the Account Settings modal and consider them as a separate business logic. So each business case had its own controller, directive and view which were consuming the RESTful API services.

4) *Implementation:*

I worked on each case one by one making sure the view (HTML) matched the directive and controller logic. This is where my experience of Teaching Assistant came in handy where I had to read code written by someone else, understand and debug errors and then implement a correct solution.

I used to implement each case test it’s working and then move ahead to the other case, if I had any roadblocks on any module I used to google for a similar problem, try to read some documentation and try a different approach, even if that didn’t work I used to stretch for help from my teammates or ask for inputs from my mentor instead of wasting time alone, I understood quickly that people there might have come across a problem sometime so it would be better of asking them, the overall communication channel in the team was very helpful. Some were JavaScript errors, some were data validation errors or SQL errors.

Although this was a challenging first project, I was able to finish this project in a timely manner and learnt a lot about how RESTful APIs work and how data moves in a big organization. What are the key areas consumers look into in an application and the importance of writing good code as it would affect so many end users.

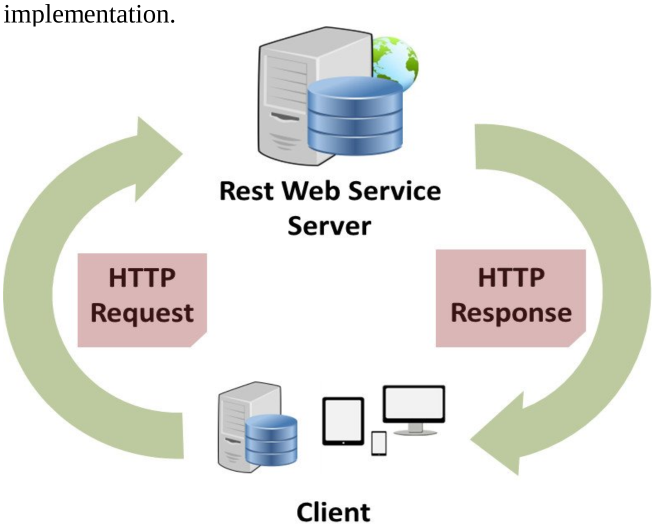
5) *Final clean up:*

I had to push most of changes from my branch into the trunk one day and it ended up breaking the build. In order to fix the build there were some test cases failing which needed to be fixed. This is where I learnt the importance of writing unit cases and them passing, I had not paid much heed to this and then we ended up fixing it and the build went through.

6) *Result:*

Every time I submitted my code for code review, if I got any recommendations for any changes or was questioned on use of a particular method of coding style I used to explain my intention or make the changes accordingly. At the end of the project I had successfully reduced the size of the code of the original file base by more than 50%. Replacing one file and breaking up into 5 respective view and controller modules. This would eventually make maintenance of code easier in the future and also provided much better readability and efficiency.

The figure below shows the basic movement of requests and response and data in a RESTful environment, the Web service contains the logic and the data for which the user requests for and this needs to be consumed by a framework, which in our case was done by AngularJS, in an MVC implementation.



VI. PROJECT III- SPRINT WORK

This was the most important part of my internship and the one which I spent most time on and the one where I learnt a learnt a lot about AGILE software development and about team collaboration. My hiring manager wanted the intern to be allocated at 100% to the scrum board where I would work on live tasks and features instead of working alone on the side and doing own projects.

The team worked on adding / enhancing the My Yield and Fertility features to give users more variables to do the calculations when they add a product to their field. There were lot of variables taken into consideration and calculations were done to predict the yield target of the field and the crop. I will explain the whole process of how the team and I went ahead working on new features in the following steps.

A. *Planning*

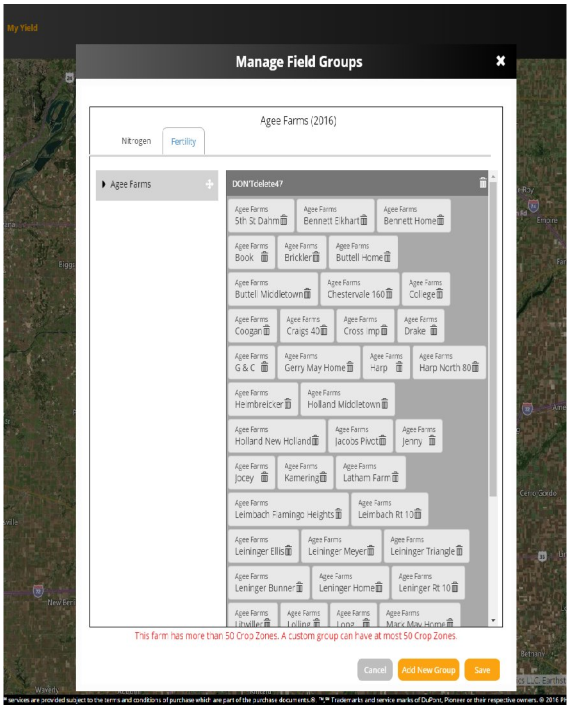
The team met every other Thursday to plan and decide what are the features (also called as stories) / bugs to work on for next two weeks (called sprints) and split them up into Front end and back end work so that the whole development team had tasks to work on. I was allocated certain features to work on during every sprint Some of the important problems were adding new features to the Fertility Add Application and implementing a drag and drop functionality to restrict the farmers from creating custom groups with just 50 farms.

B. *Execution*

We get two weeks of time to execute the stories allocated to us, as a developer I would break the story into different small tasks which when integrated would give the final product specified in the requirements document. I would split up a task into Controller changes, UI changes, Test Case writing, Service changes or DB changes whichever were necessary for that story.

I would analyze the problem at hand and go over the exact requirements of the problem and then start executing task by task. For implementing the drag and drop functionality and the Manage Groups modal I used to draw out a workflow and consult with my mentor and then start writing the code, all the farms from the left panel should be enabled to drag to the right custom group widget and then stop the action as soon as we reached 50 farms, if I ever came across any roadblocks I would evaluate an alternate option or ask for inputs from the teammates else I would wrap up the module.

One of the problems I came across was I had to make the custom drag and drop functionality work on all devices and browsers, so it was a task to make sure I considered that. I came across a similar functionality in the other parts in the application I modeled my solution based on that. The screenshot below shows the modal on which I worked on. I used to write the test cases and make sure the code I wrote for that module passed the test cases and then I went ahead and ticked off every task I completed.


Manage Field Groups

Agee Farms (2016)

This farm has more than 50 Crop Zones. A custom group can have at most 50 Crop Zones.

services are provided subject to the terms and conditions of purchase which are part of the purchase documents.®, ™,℠ Trademarks and service marks of DuPont, Pioneer or their respective owners. © 2016 PH

Another important problem I worked on was updating Stored Procedures in order to improve the SQL behavior and also make sure data validation was right and there was no duplication of columns in the result. Once there was a column being duplicated which rendered a null value on the JSON object, I worked on this problem, found the procedure and made changes to render right results so that the object could be mapped with correct values.

a) Pair Programming:

One of the tools which I learnt and implemented with my mentor was Pair programming. For some of the tasks which needed extra resources we used to pair up and start implementing code where one would review what is being written by another. One was a Driver and the other would be a navigator, switching roles frequently. This helped me pick up a lot of small nuances about development and navigating inside a vast code base, and also helped me improve my skills of problem solving as there was suggestions provided when we were coding the solution. For one of the problems my mentor wrote the Service and Directive module and I implemented the corresponding controller and view module, later I wrote the test cases and reviewed with him so that it satisfied the specifications.

For all these problems/tasks, the process followed was every morning there was a 5-minute meeting where everyone in the team got a chance to tell about what they worked on the previous day and what they were working on that day and clarify any road blocks and get inputs from the team. This helped the team track their progress and also personally I felt this was a huge motivation factor, as an Intern, if I was able to move my tasks from progress to complete; it empowered me and also provided an opportunity to showcase my skillset to the team.

C. *Complete and Deliver*

1) Retrospective

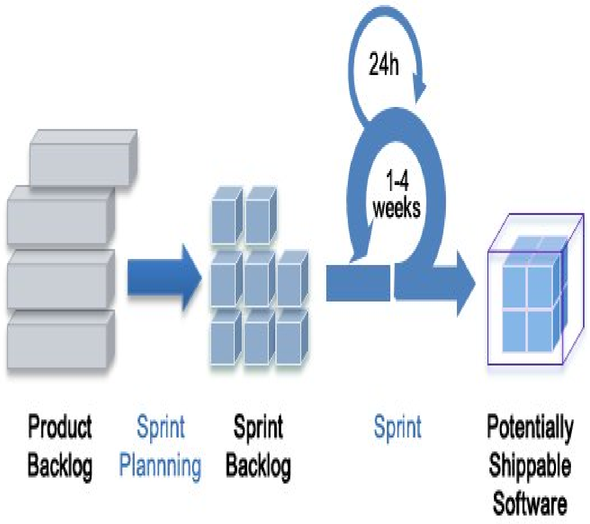
Once the planned tasks were completed the team had to present to the board the features they worked on and give them an outline about what they will be working in the next sprint. Not always the teams can complete all the tasks which they had planned to work on, sometimes there will be some cross team dependencies which might hamper the speed and affect the completion, sometime the team may have to face some critical bugs midway and dedicate resources towards that, also sometimes the team might over achieve than what they had planned, this happens when the whole team is on a sync and everything goes smoothly. All these criterions will be discussed and the team will reflect on different strategies before planning again.

2) Review

As an intern I was usually allocated reduced workload compared to the senior developers in the team, but there were occasions where if I had completed my allocated tasks, I used to get work from the backlog and finish up additional tasks which helped me understand the software better, also tested me more on my problem solving skills, and by default contributing to the team’s completion rate which thereby helped gaining the trust of the team.

I volunteered to give a presentation to the board at the end of a sprint during which, I felt I had contributed substantial amount to face an audience and answer questions. It was a nervous moment, but one I will cherish a lot, as it gave me confidence to speak to a higher audience consisting of 50 members who make business and sales decisions and helped me articulate the team’s work. This was one of the highlights of my internship.

The work flow in a sprint based environment is as shown in the figure below



VII. CONCLUSION

I was able to complete all of the tasks assigned to me which I felt was a satisfying result of my internship. I learnt a lot about the whole process of software development and how it is used to complete projects. Besides the work aspect of learning new software methodologies and techniques for writing code as a developer, I learnt how sophisticated farming field can be and how much of effort goes into the agricultural software sector. I also learned a lot of details from interaction with my team mates and higher ups. The advice and inputs I received from everyone keeps me in a good stead as I prepare to graduate and head into outside world.

This was my first internship as a software developer, so most of the things I did was a first time. I really embraced the team environment and whole process of AGILE development.

The social responsibility of a software developer and how the code written can impact so many lives made me appreciate this field I am in and also the profession. This has empowered me to be a better developer and a student.

ACKNOWLEDGMENT

Sincere thanks to the whole team at DuPont Pioneer for giving me this wonderful opportunity to spend my past summer with them. I appreciate the work ethics and team culture which was a very good motivation and made my stay enjoyable.

I would specially like to thank the hiring manager Chad Bruggemann, IT Manager for giving me this opportunity and for being an excellent teacher.

I am very thankful to my mentor Nick Anderson, Senior Software Developer for his invaluable guidance and advice during my Summer Internship.

I am also thankful to Santosh Devireddy, Technical Lead of the team for his guidance and friendly support during my stay at Pioneer.

I am also very grateful to Dr. Yi Shang, Professor, University of Missouri-Columbia and Jodie Lenser, Director of Graduate Studies for motivating and encouraging me to take up the internship.