

# Agile Embedded Systems Design Capstone Course

## The MAVRK Experience

Antonio F Mondragon-Torres, Troy Hawley, Alan Lee, and Ryan Masicci,  
Electrical Computer and Telecommunications Engineering Technology  
Rochester Institute of Technology  
Rochester, NY, USA  
afmiee@rit.edu

**Abstract—** This work is an example of a capstone senior design project in the last of a sequence of three Embedded Systems Design courses. The innovation on the presented work strives on the use of Agile project management techniques, open ended projects, ideation, market targeting, open source platforms, enhancing communications skills, and interfacing with engineers in the field. The course is very dynamic and there are just 10 weeks from product ideation to prototyping and final demonstration. The students have to demonstrate their product, upload a video demonstration, give a formal presentation before an audience, and make their design completely reusable by properly documenting and making use of version control repositories.

*Embedded Systems Design; Agile Project Management; Open Source Platforms;*

### I. INTRODUCTION

For the last of a sequence of embedded systems design courses equivalent to a senior design course, we took a non-conventional approach in which the students work as a team to create an innovative open ended project targeted to a particular market segment or market need.

What is different to a traditional course is that students were exposed to Agile Project Management (PM) methodologies and we started using an open-source platform denominated MAVRK[1] (Modular and Versatile Reference Kit) that at the time was still under development. The students were able to contribute to the development of the platform as well as to the creation of a Weather Station platform for college use. The design included an FPGA platform with a graphical user interface and touchscreen that was used to control and monitor weather variables, it communicated serially with the remote MAVRK sensing and control platform.

### II. CAPSTONE COURSE REALIZATION

#### A. Non-Conventional Approach

For the offering of this course, we have taken a different approach to what a senior capstone project traditionally is. While in a senior capstone project, a more traditional approach to project management and product development is taken, in this course the teams are structured like a small start-up company where the students will create an innovative product that will have differentiation to commercial available products and will target a specific market segment.

Another differentiation is IP reuse and collaboration across teams working on different projects. All that teams are required is to cite external consulting at the time of developing their budgets. Last but not least, we apply Agile project management techniques [2-5] which lead to innovation by allowing the students to experiment and mature their ideas. A previous version of this course was offered, but in that particular offering all teams worked competitively on an MP3 player design[2], while in this course each team choose a unique project<sup>1</sup>. A key component in this effort has been the participation of different companies as “Product owners” which help to specify, shape and collaborate during the project.

#### B. Innovation

There is a big university effort towards innovation lectures we talk about global competition and first to market constraints as well as product differentiation. We encourage component reuse in order to be more efficient and to spend time and resources on those aspects of the product that will make a differentiation factor with the competition.

#### C. Team Work

Students are encouraged to collaborate within and across teams; the only constraint imposed is to cite collaboration. In cases where students need to modify code or documentation done by previous teams in other offerings, they have to comment on the changes made and make both documentation and code available for future generations.

#### D. Open Ended

The project goals are open ended, this means that there is an initial vision of what the product will be, but the product is allowed to change direction at every “Sprint” (Sprint is equivalent to a “Milestone” in traditional project management) based on feedback from the team and the product owners/customers. At the end of each Sprint, the product should be functional to a certain degree of completion. This is a two edge sword in the sense that students could misinterpret the concept and do too little effort instead of trying to progress as much as they can at every Sprint.

#### E. Target Market

Project specifications should be focused and we do not need to reinvent the wheel every time we work on an engineering project. The market is really dynamic and those who come first

<sup>1</sup> <http://antoniofondragon.wordpress.com/student-projects-embedded-systems-design-iii-spring-2010/>

with an innovation are likely to remain the leaders. Followers need to catch up and sometimes they could vanish in this quest.

We encourage students to think a particular market that could benefit from their product. For example, it may be targeted for the masses, but that could be difficult to achieve. Students rather try to find a particular problem to be solved and attack it in an innovative way. One simple example is to offer products and services to the “Baby Boomers” generation which are currently transitioning to retirement, are wealthy and require specific services not offered before.

#### F. New Open Source Platform

As was mentioned in the introduction, students benefit to work on open source platforms since these can be assimilated easily and could start contributing to the development in a very natural and efficient form. While semiconductor companies’ technical support is tied to big customers, students do not have any influence on product definitions and thus are not considered for technical support. If a company offers an open source platform and allows global development, students could start participating immediately to the development and maturity of the platform. There are countless open source projects that are generating a tsunami in embedded systems design such as the Arduino[6], mbed[7] and BeagleBoard[8] communities to mention some of the ones that the authors are familiar with.

#### G. No User Guides or Kitchen Recipes

While students require a positive experience in a capstone course, the objective of this course is more of a reality check experience while they are still taking courses. It is very simple for students to go through instructions given in the laboratory manuals, or ask the instructor who has already solved the problem and knows the solutions. When there is no right or wrong solutions, students are exposed to real life situations, where a manager is going to ask them to perform a particular task, and in most of the cases they will require to learn the skills required to accomplish the assigned task. On the other hand a small but important number of these students could start a venture where all the innovation and ideation concepts could be applied to create the next wave of new products or services. These are some of the reasons why in this course there are no guidelines and it is valid to reuse intellectual property (IP) blocks created by former students or available as open source.

#### H. Contribution to the Platform

As mentioned before, the platforms chosen are open source and in constant evolution. Our students are involved in discovering alternative ways to use the platform according to their needs, in some cases this involves software, but in others it involves hardware. All the work performed by the students is saved in a version control repository (e.g. SubVersion), auto-documented with doxygen[9], and with instructions on how to recreate the project.

#### I. Industrial Experience

While it could seem that the open ended projects could be not very demanding, an emphasis on getting the best project achievable is given to students by encouraging them to develop professional projects that they could use to promote themselves while looking for a job. Nowadays it is very important for students to be able to push their resume to the top of the pile,

by proving to employers that they are capable of producing high quality, independent and focused products.

#### J. One to One Interaction with Engineers

One of the key outcomes of the course is on enhancing communication skills and to be able to express their ideas. While this project could be determined by students initiative, it could also solve a particular problem proposed by the “Product Owner” which could be internal (a professor) or external (an engineer at a company). It has been very important that students interact among themselves and also with external engineers already in the field and learn how to communicate and express their ideas in a concise and effective form. They should take into account that a company is a very dynamic environment and the engineers that are sponsoring the project are very busy and need to be presented with to the point information and clear requests.

#### K. Real Remote Project Development and Collaboration

For some of the projects we have been lucky that the students have been able to have remote projects kick-off meetings, as well as end of Sprint meetings to update the product owners on the status of the projects.

### III. THE PROJECT – A WEATHER STATION

#### A. Background

Students have been exposed at different levels to the Altera DE-2 board for four courses in the program. The Altera DE-2 board is an educational system with a Cyclone II FPGA running at a nominal frequency of 50 MHz. In addition to performing digital design on the FPGA using VHDL as the primary hardware description language, the previous two courses have been based on the integration of the soft core NIOS II processor to be able to create programmable Systems on a Chip (SoCs). In a previous offering of the course[2], teams worked for the first time on bringing up an LCD Touchscreen which has not been used in the past on any course. Students in this course were able to leverage on previous IP and documentation elaborated by former students. This is shown in Figure 1 where the actual GUI of the weather station is shown.

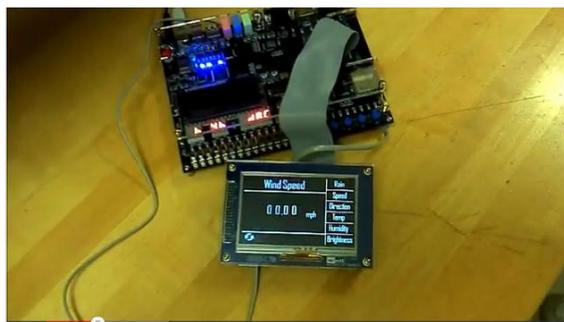


Figure 1. Altera DE-2 board used as part of the Embedded Systems Design courses with LCD touchscreen.

Students have been exposed in their sophomore year to a microcontrollers course and some of them have already been exposed to the MSP430 microcontroller used in the MAVRK platform. The MAVRK is a rapid prototyping platform which is targeted to allow engineers in the field to come up with fast implementation of their designs by using evaluation modules

(EVMs). The MAVRK platform and the actual project implementation are shown in Figure 2.

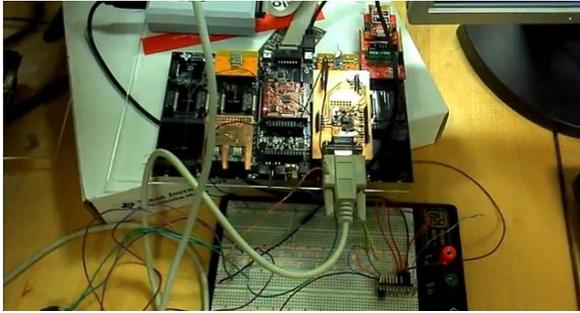


Figure 2. The MAVRK platform used at the project that is presented in this article. Additional PCB boards were designed as part of the project.

### B. Specifications

Students came with the idea to design a weather station that could be used as a platform to interface with different embedded systems. A weather station able to measure parameters such as:

- Wind speed
- Wind direction
- Rain precipitation
- Temperature
- Ambient light
- Humidity

For this project, an inexpensive weather vane was available for the team to instrument as shown in Figure 3.



Figure 3. The weather station vane used was a commercially available product and the project involved the interfacing to the MAVRK platform.

### C. Agile Project Management

While in this paper we cannot give a complete explanation of agile methodologies, we will give an example of one of the tools students used to track their project. The first stage in the project is to define a wish list with all the features they want to implement. Traditionally this can be done with post-its but in this case we used an issue tracker program called Jira<sup>2</sup> where students can list their wish lists on a planning board as seen in Figure 4. Students need to classify the tasks according to the Sprints, in our case we have 10 weeks and we divide this in two week Sprints 0.1, 0.3, 0.5, 0.8 and 1.0. The wish list become

tasks and is scheduled in a particular Sprint. Each task should be accomplishable in one Sprint. If it is predicted that a particular task will not be finished in one Sprint, it should be further divided into subtasks. This backlog of tasks will be initially be filled and will continue to be updated for the life of the project, as some tasks are given lower priorities as new tasks are discovered. Once the Sprint start, the task board shows the “To Do” tasks, the “In Progress” tasks and the “Done” tasks. This gives a very clear view of the state of the Sprint and anybody can go to the “To Do” column and self-assign a task. The task board is shown in Figure 5.

A very useful graphical tool is the “Burndown” chart that gives information on the initial number of tasks at the beginning of the Sprint, the predicted burndown rate, the number of new created tasks, the remaining values and a guideline. As can be observed in Figure 6, for this particular Sprint, students were not able to achieve their goals in terms of the number of tasks completed within the Sprint. In theory the red line and the green line should intersect at the end of the Sprint and the green has to follow the red. The orange line gives you the rate at which the tasks should be completed in one day in order to meet the burndown rate and this match the guideline. We can also observe that the blue line indicated that new tasks were added in the middle of the Sprint. This should not have happened, but it means that students discovered something that they did not realized before and then had to be included.

Does this means that Agile project management methodology does not work? No, it means that the students are been exposed to this methodology for the first time and also they are working on their own for their first time by defining and creating their own tasks. This means that they do not have the skills required to predict accurately the time duration of particular tasks and also that they could not define perfectly all the tasks to be done. These situations are a great learning experience and helps with documenting and keeping track of team activities.

### D. Hardware

In this particular case the platforms to be interfaced would be the DE-2 FPGA board with a LCD touch-screen as the user interface and control and the MAVRK motherboard and EVM modules.

The DE-2 platform will be in charge of the user interface and light control. For this purpose Altera SOPC builder was used to create the required hardware to control the LCD touch-screen, a simulated home light dimmer and the serial communication interface with the MAVRK. A Nios II processor was instantiated and programmed to accomplish these tasks. Additional hardware components were created in VHDL and added as custom components to the NIOS II system

The MAVRK platform is in charge of all measurement and control functions, both analog and digital. The boards will communicate through an asynchronous serial RS-232 communications link and a protocol with a simple packet structure was devised to accomplish this task. Asynchronous serial communications were chosen in order for the students to realize their own rules and finite state machines to handle inter

<sup>2</sup> <http://www.atlassian.com/>

platform communications. There are some other communication standards and protocols that could have been used such as I2C and SPI, but in this case we wanted students to realize the efforts required to communicate two asynchronous devices. In addition to this decision, all embedded systems usually have this type of interface either as a direct link or as a backdoor for debugging. Moreover, a printed circuit board was designed on-site to create an RS-232 interface to the MAVRK platform. This exposed students to learn on how to adapt their ideas to the selected platform.

#### E. Software

Programs for the NIOS II processor and for the MAVRK were all done in embedded C language. High reutilization was achieved in both platforms by leveraging on previous projects for the Altera platform and on libraries available on the MAVRK platform. This allowed the students to focus on the differentiation on their products by creating a more advanced graphical user interface and to communicate the platforms using serial communications.

#### F. Prototyping

##### 1) Common Platform

As mentioned before, students have been exposed for four quarters to the DE-2 and are comfortable designing with this platform. The constraints are that this platform does not contain any available analog interfacing ports that could be used to extract information from the weather sensors. Also the touch screen is to be installed on a convenient place for the user.

##### 2) Open Source Platform

The MAVRK platform has a series of slots that allow different types of components to be integrated. Following a traditional analog processing signal chain these are:

- Signal Conditioning (SC) slots
- Analog Front End (AFE) slots
- Radio Frequency (RF) slots

The MAVRK is based on the MSP430 and in particular, a MSP430F5438A is used as the main processor. Each slot has a combination of analog signals, digital signals and communication interfaces. So in addition to the traditional analog signal processing done on senior projects, the students are exposed to serial protocols and interfaces such as:

- Serial Parallel Interface (SPI)
- Inter-Integrated Communications I<sup>2</sup>C bus
- Universal Asynchronous Receiver and Transmitter (UART)

##### 3) Putting all together

The Agile methodology calls for self-organization. One student specialized in the MAVRK platform, both hardware/software and serial communications. Another student on the graphical user interface (GUI) and serial DE-2 serial communications, while the other student was more on the hardware components to be interfaced to the NIOS II system.

Even that each student had a different specialization; the three students were collaborating where it was needed, thus

maximizing their time and achieving better results at the end of each Sprint.

#### G. Demonstration and Dissemination

In order for students to get a passing grade in the course, the students have to:

- 1) Give an *In Situ* demonstration of their working project before their final presentation to an audience of students, faculty, and family.
- 2) Upload a YouTube video with a presentation of their final product.
- 3) Present their work formally to an audience using PowerPoint and the video uploaded. In this case several engineers who were "Product Owners" joined the presentations remotely and participated in the Q&A.
- 4) Prepare documentation on how their project could be reproduced and lead the course instructor on how to reproduce their project based solely on documentation and code available on the version control repository.

## IV. RESULTS

In particular, this project was very successful and the outcomes were recognized by Texas Instruments in their internal communications and external communications. Students were very motivated to see all their work recognized and prepared them for the workforce by applying the workable skills obtained through the program that led them succeed in this project. The final presentation and YouTube video are available at:

<http://www.youtube.com/watch?v=Sa7ykLBKMT4>  
<http://www.slideshare.net/afmiee/weather-meter>

## V. LESSONS LEARNED

#### A. Creativity and work under pressure

Students are capable of being creative at the same time dealing with time pressure due to the finite length of the course. In this particular course there are just 10 weeks available from idea to demonstration and that is the reason component and IP reuse is the key to succeed. Students can concentrate on the main product differentiation by gathering usable blocks.

#### B. Professional Communication

Students were able to communicate effectively with design engineers in the field to accomplish their goals that were done through remote meetings, email and version controlled repositories.

#### C. Reality Check

While we present here that everything is perfect and all students are motivated, there have been other teams that were not able to organize during the 10 available weeks or that faced road blocks that were not being able to overcome. Traditionally throughout the program, students are given laboratory exercises which guide them step by step and let them know what the desired results are. In courses like the presented here some students could be overwhelmed by not having the experience to take decisions by themselves, to propose innovative ways to perform the tasks or how to organize among themselves. In this

case as mentioned Agile PM methodologies can be a double edged sword where students do not know even how to begin.

## VI. CONCLUSIONS AND FUTURE WORK

By using a new platform, there are no user guides, kitchen recipes or cookie cutter laboratory exercises; all development is new and limited by student's initiative and engagement. In this particular experience, a one to one interaction with real development engineers had to be performed in order to bring the project to a reasonable state of completion.

The students enjoyed the interaction with their industrial colleagues, contributed to the platform development and were exposed to a real remote project development and collaboration. The lessons learned from this open ended project are that: students are capable of being creative at the same time dealing with time pressure due to the finite length of the course, and are able to communicate effectively with design engineers in the field. Students made a formal presentation of their project, give live demonstrations as well as prepared and edited a video for further dissemination and to set precedence on their experience. The company used this video internally to demonstrate the platform capabilities.

Based on this work a new project has been proposed and students are currently working on the wireless implementation of the weather station by leveraging on the wireless communications between the main MAVRK board and a smaller  $\mu$ MARK board with RF capabilities. It is intended that the new team will install permanently the weather station and this will allow us to perform continued interfacing with the platform for future projects or courses.

## ACKNOWLEDGMENT

This work is the result of the support from multiple people and multiple companies who have supported the course. We

would like to acknowledge Texas Instruments MAVRK team for providing the resources both material and human to be able to participate in the early stages of MAVRK development. We would like to acknowledge Altera University program that has been supporting this course by providing hardware and software components to expose students to advanced FPGA design. We would like to thank Atlassian for granting the licenses for their Agile project management suite which made this course possible. We would also like to thank the ECTET department at RIT for providing the computer and financial resources to build the product.

## REFERENCES

- [1] Texas-Instruments. (2012, 04/22/2012). MAVRK Introduction - Texas Instruments Embedded Processors Wiki. Available: [http://processors.wiki.ti.com/index.php/MAVRK\\_Introduction](http://processors.wiki.ti.com/index.php/MAVRK_Introduction)
- [2] A. F. Mondragon-Torres, A. Kozitsky, C. Bundick, E. Mc Kenna Jr., E. Alley, M. Lloyd, P. Stanley, and R. Lane, "Work in progress: An Agile Embedded Systems Capstone Course," presented at the 2011 Frontiers in Education, Rapid City, SD 2011.
- [3] J. A. Highsmith, "Agile project management creating innovative products / Jim Highsmith," ed: Upper Saddle River, NJ Addison-Wesley, c2010.
- [4] C. Larman, Agile and Iterative Development: A Manager's Guide: Addison-Wesley Professional, 2003.
- [5] R. Goering. ( 2010). Can Agile Software Development Methods help SoC design? Available: <http://www.cadence.com/Community/blogs/ii/archive/2010/08/18/can-agile-software-development-methods-help-soc-design.aspx>
- [6] (2012). Arduino - HomePage. Available: <http://www.arduino.cc/>
- [7] (2012). Rapid Prototyping for Microcontrollers | mbed. Available: <http://mbed.org/>
- [8] (2012, 04/22/2012). BeagleBoard.org - default. Available: <http://beagleboard.org/>
- [9] (2012, 04/22/2012). Doxygen. Available: <http://www.stack.nl/~dimitri/doxygen/manual.html>

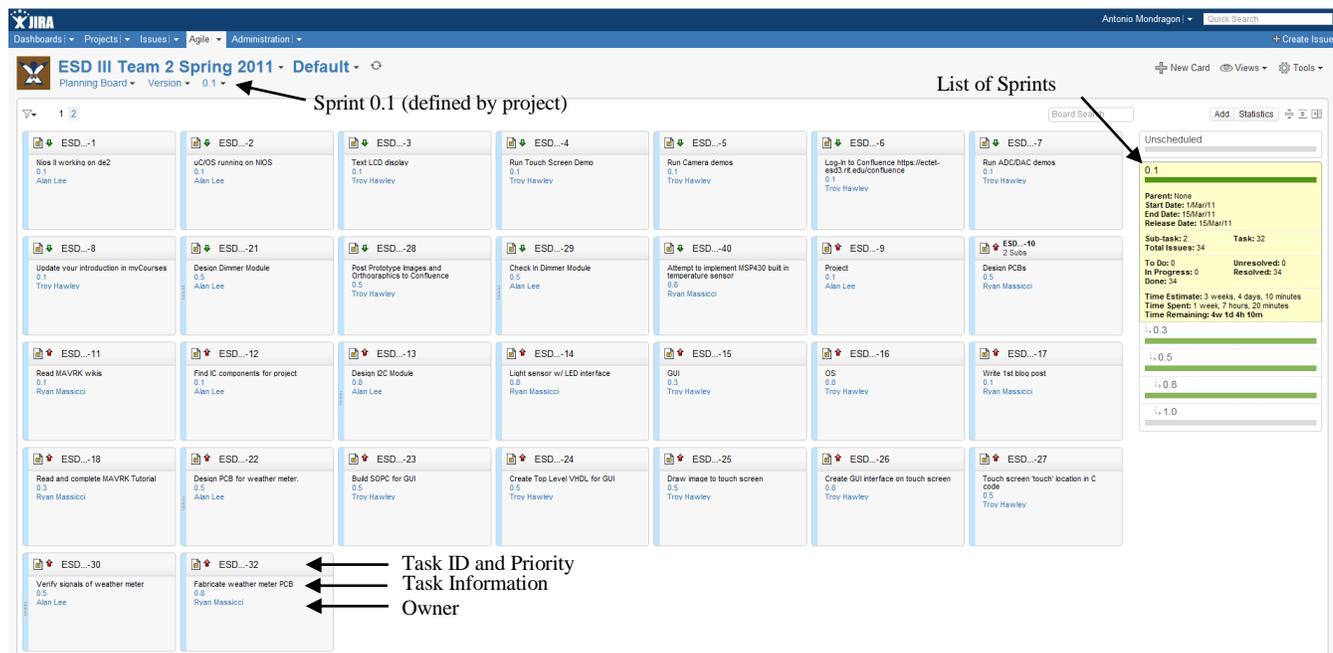


Figure 4. Jira planning board. All the tasks are entered in this view. At each Sprint tasks are prioritized and scheduled. If a task cannot be completed in one Sprint, then it should be divided into subtasks.

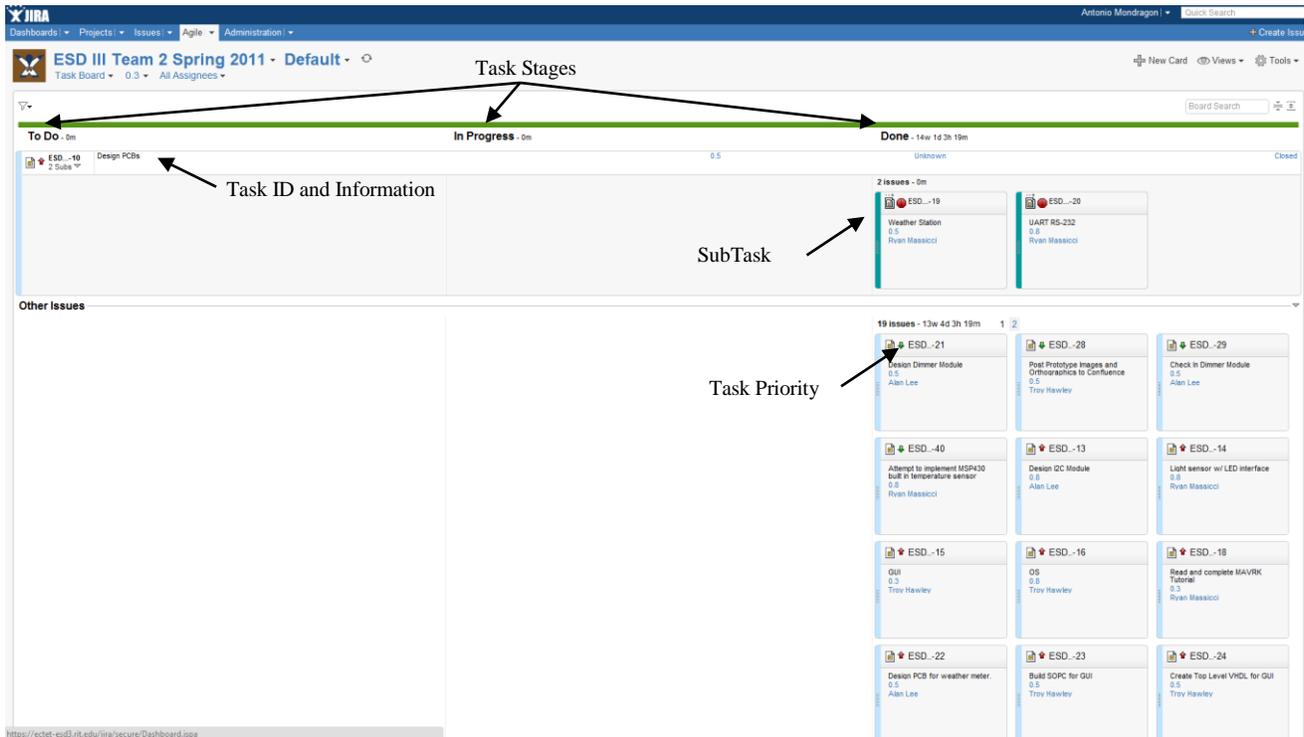


Figure 5. Jira Task Board example. As can be observed, all tasks for Sprint 0.3 have been completed. Also it can be observed that the task "Design PCBs" (upper left corner) was subdivided into two tasks. The tasks were assigned priorities previous to scheduling (represented by the arrow icons on each task).

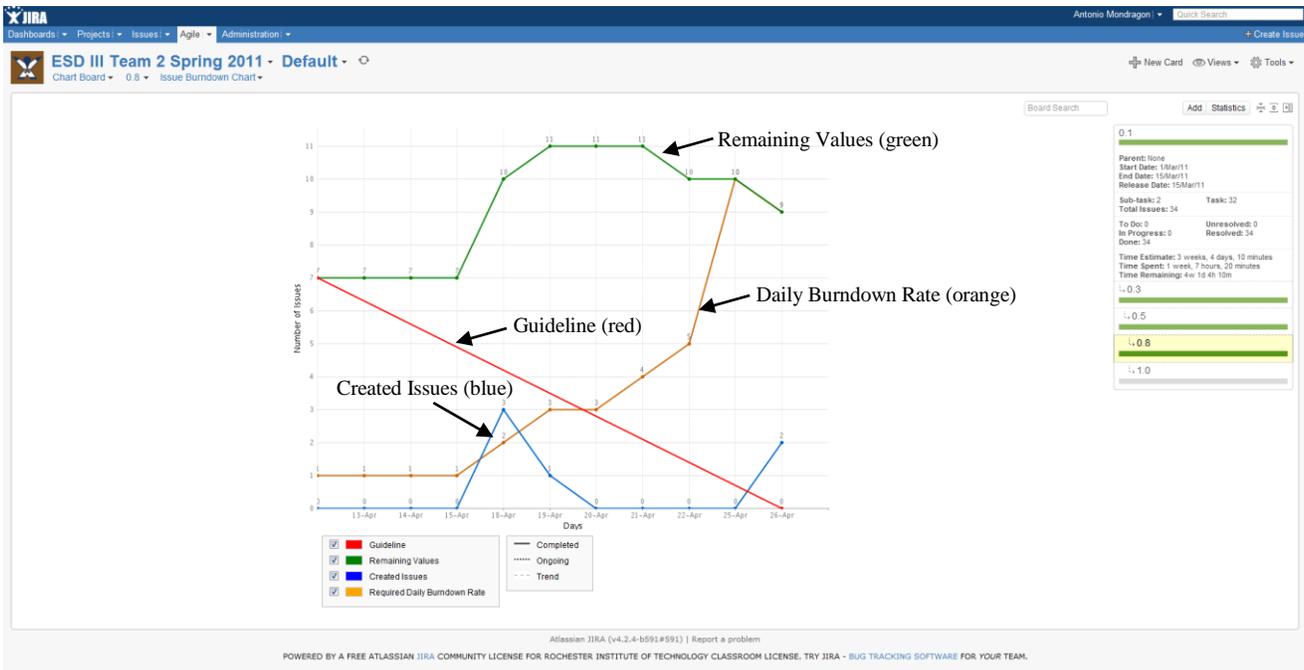


Figure 6. Jira issue burndown chart for Sprint 0.8. The green line represents the tasks in progress. The orange line represents the burndown rate, which is the number of tasks to be finish each day in order to meet the guideline (red line). The blue line represents the newly created tasks per day (In theory it should be zero).