**Graduate Projects**

University of Colorado at Boulder

Aerospace Engineering Sciences

ASEN 5018/6028 –Spring 2015

|  |
| --- |
| **Drones Versus Zombies (DVZ)**  **Project Management (PM) /Systems Engineering (SE)**  **Best Practices and Lessons Learned** |

|  |  |  |
| --- | --- | --- |
| **Approvers List** | | |
|  | Title | Name |
| Prepared By | Project Manager | Garrett Brown |
|  | Systems Engineer | Edward Scott |
|  |  |  |
| Approved By | Systems Engineer | Edward Scott |
| Approved By | Project Manager | Garrett Brown |

# Introduction & Summary

The DVZ Systems Engineer serves as the central point of communication between all technical subsystems and is primarily responsible for managing the system design. The Systems Engineer is responsible for determining, managing, and verifying all levels of the requirements baseline, and is responsible for ensuring that the as-built system satisfies the requirements.

The DVZ Project Manager serves as a central point of contact between all subsystems and team members and between the team, advisor, and customer. Additionally, the Project Manager is responsible for scheduling and coordinating team activities, directing team workflow, ensuring the completion of tasking, managing the team budget, and managing the configuration of all team documentation.

# Lessons Learned

## Systems Engineering

### Tasking

* Make sure everyone on the team has weekly tasks assigned to them with clear, quantifiable, goals and status those goals at each meeting to ensure team members stay on track.
* Keep up-to-date with what is being done and when it is planned to be done so technical development stays within the design. Because DVZ uses a more dynamic development approach than some projects, in-depth monitoring of progress is necessary to prevent significant scope creep and ensure integration will be possible.
* Encourage team members to accomplish tasks by leveraging as much existing work as possible in order to streamline development and make the project simpler.

### Meetings

* Come to meetings with clear status regarding what has been done and what still needs to be done.
* Make sure to ask everyone what they are concerned about every meeting, people will rarely express their concerns unless asked. Additionally, track concerns from a systems perspective between meetings to ensure that ongoing work serves to address team concerns
* Attend as many subsystem-specific meetings as possible

### Customer/Advisor Handling

* Be precise when talking to Dr. Frew. He will get bogged down in semantics if given the chance, so be clear and concise when talking with him.
* Don’t necessarily ask the customer when you have already made a decision, lead him to the same conclusion you’ve already made and justify the decision. The DVZ customer is less concerned with specifying design details and more concerned with a robust finished product.

### Class Requirements vs. Project Requirements

* Requirements matter to a point. You want to build the right thing and therefore need to define what that thing is but you don’t want to restrict your design space. Additionally, requirements need to remain flexible to account for the use of off the shelf products which may not meet stringent performance specifications.
* The class does not require much from an academic standpoint, so spend more time focusing on ensure that the system being created is robust and functional. Product functionality is a key aspect of DVZ, thus don’t get overly caught up in management approaches.
* Despite a more product-centric focus than some larger teems, keep future teams in mind when designing. While the customer’s primary concern is not requirements and documentation, both are necessary to maintain effective project continuity.

## Project Management

### Customer Interaction

* Ensure the Customer and Advisor input is accounted for, but is weighed appropriately against team decisions. Remember that the DVZ project is academic in nature and can accommodate a greater amount of trial and error than a more rigorously defined project.
* Establish a project baseline as early as possible and adhere to it as closely as possible.
* When possible, solicit customer input in written format to prevent scope creep and provide a starting point for future discussions.

### Team Communication

* Send important emails more than once and highlight key information.
* When a reply is necessary, include [response required] in the subject line to ensure that team members are clearly informed that you expect input
* Send weekly reminders for standing meetings even if meetings are recurring to ensure everyone knows what the schedule for the week is.
* Send tasking direction and short-term timelines in textual format so team members can review them as necessary. Ensure that tasks are clearly tied to responsible team members and to expected completion dates so there is no room for misinterpretation.

### Task Coordination and Scheduling

* Flow the WBS directly from requirements and architecture but leave space for unexpected tasks. Due to the relative technical inexperience of the team members and the complexity of off-the-shelf hardware, tasks may arise which could not have been anticipated (e.g. Chi Squared distribution analysis during Phase 2)
* Track team progress and tasking at regular intervals to ensure that team members are working toward appropriate goals.
* Ensure that team member tasking falls within the confines of team member roles. The DVZ team has historically been filled with great people driven toward success, but that drive sometimes leads to unneeded tangents, so keep the team on track.
* Keep an up-to-date list of ongoing tasking with due dates to ensure team members have easy access to what tasks must be completed and when.

### Configuration Management

* Provide a common, easily navigable repository for team data to ensure that team members access correct information. Maintain the repository on a regular basis and remove or archive old information.
* Track document updates for historical reasons and keep old versions.
* Create templates for recurring work to save time and ensure quality.

### Financial

* Create a single point of contact for purchasing to ensure that all purchases are approved and meet the budget.
* In the future, it is suggested that the project manager be the POC for purchasing to ensure that all purchasing decisions are flowed to the top. Budget should be guarded carefully.

# Role Continuity

## Systems Engineering

### Continuity Documentation

* As you might be unfortunate enough to find out, continuity documentation is difficult to maintain. Task someone at the start of the year with keeping records of all tests/results/reasons for changes.
* Keep scalability of the design in mind. For example if your year is focusing on controllers you don’t want to design a solution that will hinder object recognition in later years.
* Start using the git repository that was setup and move away from using Dropbox for a code repository. This was a stupid thing of us to do and there were numerous times we regretted it.

### The SE’s Job

The Systems Engineer needs to pay attention to requirements. You will inherit fully defined level 1 requirements and extensively flushed out level 2 requirements. The idea behind the purpose of these is that the level 1 requirements should encompass all design space that the customer and advisor want you to even consider. It specifies things like team of vehicles, multirotor platforms, etc. Ideally you will not need to change any of these requirements even if major changes are made to the baseline design as it stands. The level 2 requirements are more specific to the baseline design as it currently stands. Things such as ROS architecture, wifi communication, etc are encompassed by this set. While changes might be made at the level 2 requirements, there should be plenty of room to flesh out any design changes in the level 3 requirements which have been paid much less consideration.

The Systems Engineer should also have great knowledge of their personnel. You should get to know your team well and have knowledge of their schedules, what motivates them, who will have good ideas on how to proceed, and who easily wanders off track. The team size of DVZ is tracked to be <10 as of May 4th 2015, so it should be fairly easy to know everyone on a personal level and have a good working relation with them.

The SE should have a constant working knowledge of the system. At no point should you be unaware of the current planned solution for any problem. If you find yourself lacking knowledge of how any piece works immediately setup a meeting with that subsystem lead. Note, you don’t need to know all the nitty-gritty details but certainly the system level solution. For example you should know if you are using an Extended Kalman Filter vs and Uncented Kalman Filter but you don’t necessarily need to know the equations behind whichever is being used (Though if you can you should know that too). Ideally if any of your technical leads will be gone on travel you should be able to fill the void given only a few days of catch-up detail rather than have to wait for them to return.

Finally, the SE should try as hard as possible to be present for every test. This includes everything from full systems integration to subsystem component checkout. Take note of what the test is and why it is being done. After the test is concluded either write, or task someone to write, a brief summary email to the team or fill out a formal test report. While it’s a pain in the ass to do it will benefit you later to have these documents. Work very closely with your test engineer to ensure that someone is documenting testing progress

### Starting Points

1. Review current design solution with respect to ability to meet all level 1 requirements
2. Continue flowing level 2 requirements into quantifiably testable level 3 requirements
3. The current design solution has many limitations with respect to the level 1 requirements. First take a look at the level 1 requirements at /DVZ/Systems Engineering/Requirements Specifications/DVZ Requirements (L1). Then review the design and how it may or may not meet each requirement with how it stands so far. The most helpful team input on this matter can be found in the Spring 2015 final presentation in /DVZ/Project Management/Configuration Management/Formal Presentations/DVZ-PM-011 DVZ Phase 2 SFSR Charts/Team Member Inputs/ DVZ-PM-011 DVZ Phase 2 SFSR Charts\_Eddy. Specifically look at the Requirements slides and the proposed architecture changes.
4. Start thinking about how good is ‘good-enough.’ Dr. Frew and Dr. Ahmed are not the best at quantifying these numbers so begin by looking at the testing environment and what physical tolerances will be necessary to operate safely there. This will help in telling people that they need to stop developing one design solution and begin work on another.
5. The level 2 requirements are flowed down with our design decisions in mind. If a requirement no longer makes sense for the scope of the project don’t be afraid to remove it.
6. It is assumed that flight testing can be done in parallel with system design. It may be necessary to descope the project to allow for more rigorous designing and push less hard for actual flight testing.

### Document Locations

* General Systems engineering: /Dropbox/DVZ/Systems Engineering
* Level 1 requirements can be found in: /Dropbox/DVZ/Systems Engineering/Requirements Specifications/DVZ Requirements (L1)
* Level 2 requirements can be found in: /Dropbox/DVZ/Systems Engineering/Requirements Specifications/Element Requirements (L2)
* Level 3 requirements can be found in: /Dropbox/DVZ/Systems Engineering/Requirements Specifications/Subsystem Requirements (L3)
* Con-Ops and FBD: /Dropbox/DVZ/Systems Engineering/DVZ CONOPS and FBD

## Project Management

### Initial Steps

Due to the unique relationship between the DVZ customer and advisor, it is suggested that the PM and SE meet with the customer and advisor before carrying out significant tasking direction or project planning to ensure that goals and expectations are clearly understood and documented. After initial meetings with the customer and advisor, it is suggested that the PM and SE work with the customer to receive approval on a top level requirements set and provide a common baseline for additional development. After a requirements baseline is established and architecture is created, the PM should flow a Work Breakdown Structure (WBS), which, though not necessary for customer expectations, will provide a set tasking list for the semester. Due to the relatively undefined nature of the DVZ project, a WBS and related Integrated Master Schedule (IMS) is highly suggested to ensure that day-to-day tasking meets a necessary goal and to help prevent unnecessary tasking. After the initial customer interaction, it is suggested that a standing customer meeting be scheduled without the advisor.

### General Management Strategies

For scheduling and planning, the Waterfall method is NOT recommended because it is too inflexible for the research-level focus placed on the DVZ project. To allow for a more dynamic development environment, a modified Agile method is suggested which accounts for an overarching schedule, and, more importantly, the implementation of requirements rather than stories. Due to the high potential for requirements volatility and scope creep experienced in the past with the customer/advisor combination, and to allow for a better-documented baseline over a longer duration, a preset requirements baseline is highly recommended, however, the DVZ project is research oriented and can accept a greater amount of scope modification than a more traditional project, thus, don’t be afraid to rescope or re-evaluate the schedule if necessary.

### Timecards

Timecards are generally due weekly on Mondays and cover a seven day span from the previous Saturday to the Friday preceding the due date. Timecard templates will be provided as part of the course material, but can be found at the Timecard directory given below. It is suggested that the Project Manager gather and compile the Timecards on a weekly basis. Note that while the Project Manager should remind team members to create timecards, the compiled timecard may be turned in with input from team members missing.

### Team Status

It is suggested that team status be gathered at least every other day either via email, or during meetings. Generally, one meeting a week includes the advisor, and it is suggested that the advisor meeting be used to compile all status for the week and generate short term tasks for the next week. Including the advisor during the tasking process allows external input and ensures that all tasking is necessary. It is suggested that all progress and tasking be recorded in some way; the current format is presentational with Quad Charts and a running actions list. Past status presentations can be found at the location provided below:

* DVZ/Project Management/Internal Status

### Configuration Management

The DVZ Configuration Management (CM) folder is provided below and should be used to house all significant (numbered) documentation with the exception of Test, SE, and Project Management artifacts which may reside in their respective folders for ease of access. All DVZ deliverables and critical documentation should be kept in CM and should be provided a document number. Existing DVZ document numbers are recorded in the DVZ-CM-001 DVZ Document Listing.xlsx which is in the top level of the CM folder. Additionally, the DVZ Document Listing contains descriptions and locations of all numbered DVZ documents. When significant updates are made to a numbered DVZ artifact (generally in preparation for a milestone or due to redesign), the associated artifact should be given an updated revision marker. Revision markers should be appended to the end of the document title and should begin with Rev (Revision) – for newly released documents before proceeding with Rev A, Rev B, and so on. The entire DVZ directory was archived and provided to the PM for Fall 2015, and it should be loaded as-is into either the git-hub or Google Drive to allow for ease of collaboration. Ensure that all team members are aware of and have access to the DVZ directory.

### Fall 2015 Specific

* Because so little of the team is returning, spend at least the first two weeks having team members review all documentation from the 2014-2015 school year so they have a solid notion of the project scope and current state
* Before beginning any work, define clear roles to prevent overlapping work
* While the team reviews documentation, define a clear timeline with set dates which can be used to drive tasking. The lack of a hard timeline was an issue in the 2014-2015 school year and led to some inefficiency, thus, define a timeline and milestone presentations early and establish a schedule.
* Suggested initial tasks are as follows:
* Project Manager: Generate a WBS based on the expected scope for the semester and use the WBS to create a timeline at 90% confidence before the end of week two. Generate a schedule and ensure that it is flowed to the team. Schedule meetings including advisor and customer meetings. Get a handle on timecards, status, and weekly tasking directions.
* SE: Gain familiarity with the existing requirements and update them as necessary. Review the current status of the requirements per the design document. Finalize requirement flowdown and update the architecture accordingly.
* Test: Review test procedures and results from Spring 2015 and update procedures based on Fall 2015 scope. Begin planning key tests. Characterize any new hardware ASAP and determine what it can and cannot be used for based on its inherent accuracy.
* HWE: Ensure the X8 quads can be flown, then move to begin downsizing the system to a new quad. This may require the purchase of a smaller platform, but talk with James Mack first about using the platform he bought for this purpose in Spring 2015
* CommS: Resolve wifi issues in hallway and implement a robust solution for testing outside of the RECUV lab as quickly as possible. Move on to implementing communication for multiple platforms. As personnel allows, finalize the Gazebo model
* LAMSS: Exercise current Kalman Filter, then begin looking at better options for acceleration measurement and Odometry including but not limited to standalone accelerometers and visual odometry. After Localization is more robust, begin looking into object identification.
* MS: Exercise current controllers, then begin looking at new autopilots that allow faster sampling rates. Move on to tuning the Yaw controller to allow for easier movement into path planning and search algorithms. As personnel allows, finalize system models in MATLAB (or other language) and leverage them to begin investigating path planning for one quad and multiple quads.
* Facilitate the following initial purchases or ensure that sufficient hardware has already been procured
* Blades for the X8+
* All hardware for the X8+
* Spare bladeguards
* Spare cables
* New smaller quad if necessary
* New higher resolution Hokuyo if necessary
* Standalone accelerometers if necessary
* Cameras for vision based Odom and object identification

### Documentation Locations

Prior Project Management documentation and methods can be found in the following locations on the Dropbox:

* General Project Management Documentation DVZ/Project Management
* DVZ Configuration Management: DVZ/Project Management/Configuration Management
* DVZ Work Breakdown Structures: DVZ/Project Management/Work Breakdown Structure
* DVZ Financial Tracker: DVZ/Project Management/Financial/Expenses
* Continuity Documents: DVZ/Project Management/Subsystem Summary
* Timecards: DVZ/Project Management/Timecards
* Internal Status: DVZ/Project Management/Internal Status
* Document, Presentation, and other templates: DVZ/Project Management/Configuration Management/Templates