Systems Engineering and the V-Model: Lessons from an Autonomous Solar Powered Hydrofoil

Joshua Sutherland[1], **Haruya Kamiyama**[1], **Kazuhiro Aoyama**[1] **and Kazuya Oizumi**[1]

ABSTRACT

The development activities performed by a student team to create and race an autonomous solar powered hydrofoil are described. These development activities are analyzed by way of four V-Model views and Systems Engineering best practices. Subsequently an alternative development strategy is proposed. By way of this example, insight into the challenges of systems development and how formal Systems Engineering tools can improve success likelihood is provided.

KEY WORDS

Systems Engineering; V-Model; Lean Systems Engineering; Engineering Education; International collaboration

INTRODUCTION

In this paper the product development activities performed to create an autonomous solar powered hydro foiling boat are decomposed by way of the V-Model views presented by (Scheithauer & Forsberg, 2013) and Systems Engineering best practice and enablers presented by (Oppenheim, Murman, & Secor, 2011).

As described by (Oppenheim et al., 2011) “Systems Engineering (SE) is regarded as a sound practice but not always delivered effectively, as documented in recent NASA [National Aeronautics and Space Administration], GAO [Government Accountability Office], and DoD [Department of Defense] studies.” (Oppenheim et al., 2011) by way of presenting data from various US space and defense development programs indicate that government product development programs often have large inefficiencies, with charged time being estimated as 60-90% waste.

The aim of this paper is to explore the deficiencies in the product development activities adopted on a student project such that lessons can be learned and a better process adopted for future developments. Further, deficiencies with the Systems Engineering process can be identified such that future research directions can be proposed.

SYSTEMS ENGINEERING AND THE SOLERBOAT

Systems Engineering

As per its industry body Systems Engineering is defined to be “an *interdisciplinary approach and means* to enable the *realization of successful systems*. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while *considering the complete problem*: Operations, Cost & Schedule, Performance, Training & Support, Test, Disposal and Manufacturing.” (INCOSE, 2004). Further “*Systems Engineering integrates all the disciplines and specialty groups* into a team effort forming a *structured development process that proceeds from concept to production to operation*. Systems Engineering considers *both the business and the technical needs* of all customers with the goal of providing a quality product that meets the user needs.” Italics (added by the authors), emphasize the importance placed the integration of multiple disciplines to consider the complete problem and deliver a solution which solves it. As per these definitions Systems Engineering is attempting to be the

[1] Department of Systems Innovation, Graduate School of Engineering, The University of Tokyo, Japan

discipline which enables the development and deployment of systems utilizing expertise from multiple domains, to be successful.

SolarBoat Racing Challenge

Late every August, for one weekend, Lake Biwa in Japan hosts an inter university solar powered boat racing competition. Teams each bring boats which they have developed and are challenged to race them automatously over a 20km course powered by using a maximum of $2m^2$ of solar panels and 20Wh of lead based batteries. Points are awarded for time taken to complete the course and adherence to the course waypoints. Each team follows their boat from a support vessel and can make repairs and modifications in the event of breakdowns, however such activity incurs a penalty.

Such a challenge fits with the definition of what the Systems Engineering discipline aims to support in descriptions provided by (INCOSE) described in detail earlier, in that the challenge:

- Is interdisciplinary involving: Mechanical engineering, Electrical engineering, Software engineering, Control engineering, Structural engineering, Naval architecture and Automatous vehicles.
- Covers a full product lifecycle: Concept -> Detailed design -> Manufacture -> Assemble -> Test -> Operate
- Requires the management of competing needs: Winning vs. Learning vs. Costs

Development and Race in 2014

Following on from collaboration in 2013, in the 2014 event the University of Tokyo, Department of Systems Innovation formed a collaborative team with KTH Royal Institute of Technology, Sweden. The combined team consisted of twelve masters' students from the University of Tokyo and nine from KTH. For the KTH students the project was spread over the entirety of 2014 with the race in Japan acting as a 3/4 point mile stone. For the Tokyo students the program had time scales which were significantly shorter with the project kickoff occurring in early April and ending at the race. A high level summary of activities is shown in Figure 1. The focus of this paper is upon how development occurred in Japan by the Tokyo students.

With the aim to more successfully aid communication and allocate work both KTH students and Tokyo students arranged themselves into three sub teams: Control/Navigation, Structures and Propulsion/Powertrain. Each being tasked to design and

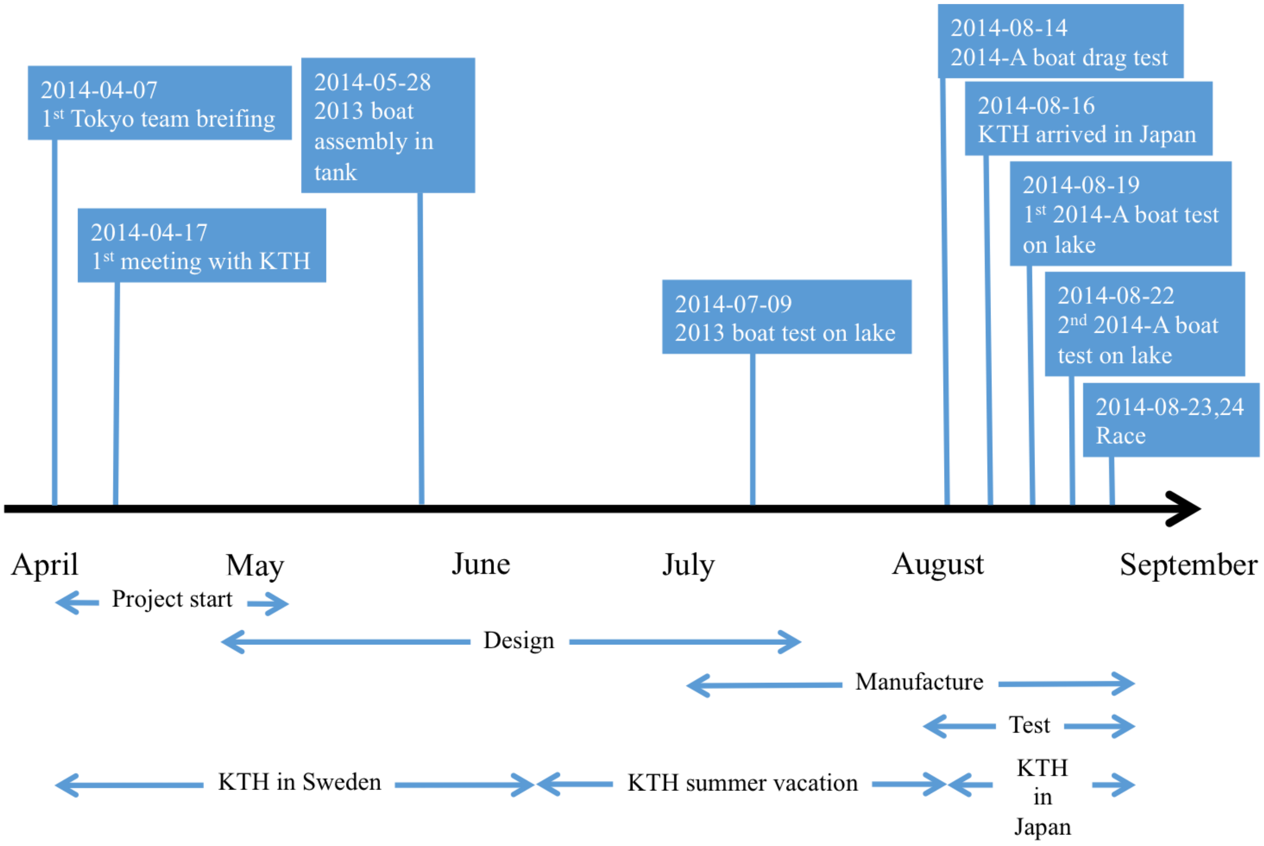

Figure 1: Simple summary of milestones completed by the Tokyo team

manufacture respective sub systems. As shown in Figure 1, KTH students were to be in Japan for only one week before the race, prior to this all communications with them was via email and video conferencing. Further KTH summer vacation was during a large section of the Tokyo development time.

Two hydro foiling boats (2014-A and 2014-B) were designed, built and raced, ultimately ranking 3rd and 4th place, suffering multiple breakdowns and failing to complete the course, being beaten by significantly simpler designs using displacement hulls. The following figures are provided to give the reader an overview of the craft constructed: Figure 2, shows photos of the boats in operation foiling at around 2ms^{-1}; Figure 3, a general overview of boat 2014-A's design and dimensions and Figure A1 (in Appendix A) a component structural decomposition hierarchy of 2014-A. Both craft were primarily manufactured from Carbon Fiber Reinforced Polymer (CFRP) and had masses of 14kg (2014-A) and 16kg (2014-B).

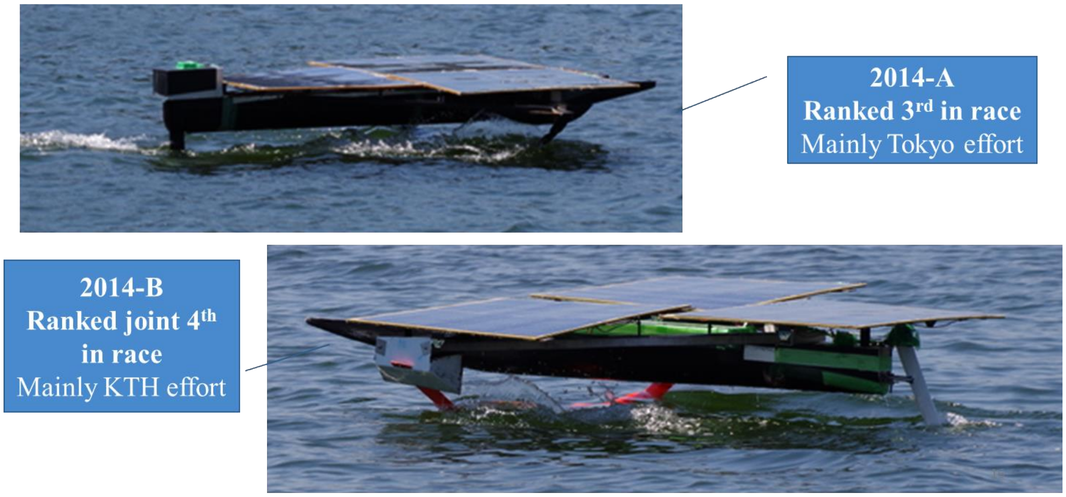

Figure 2: Completed boats foiling under solar power

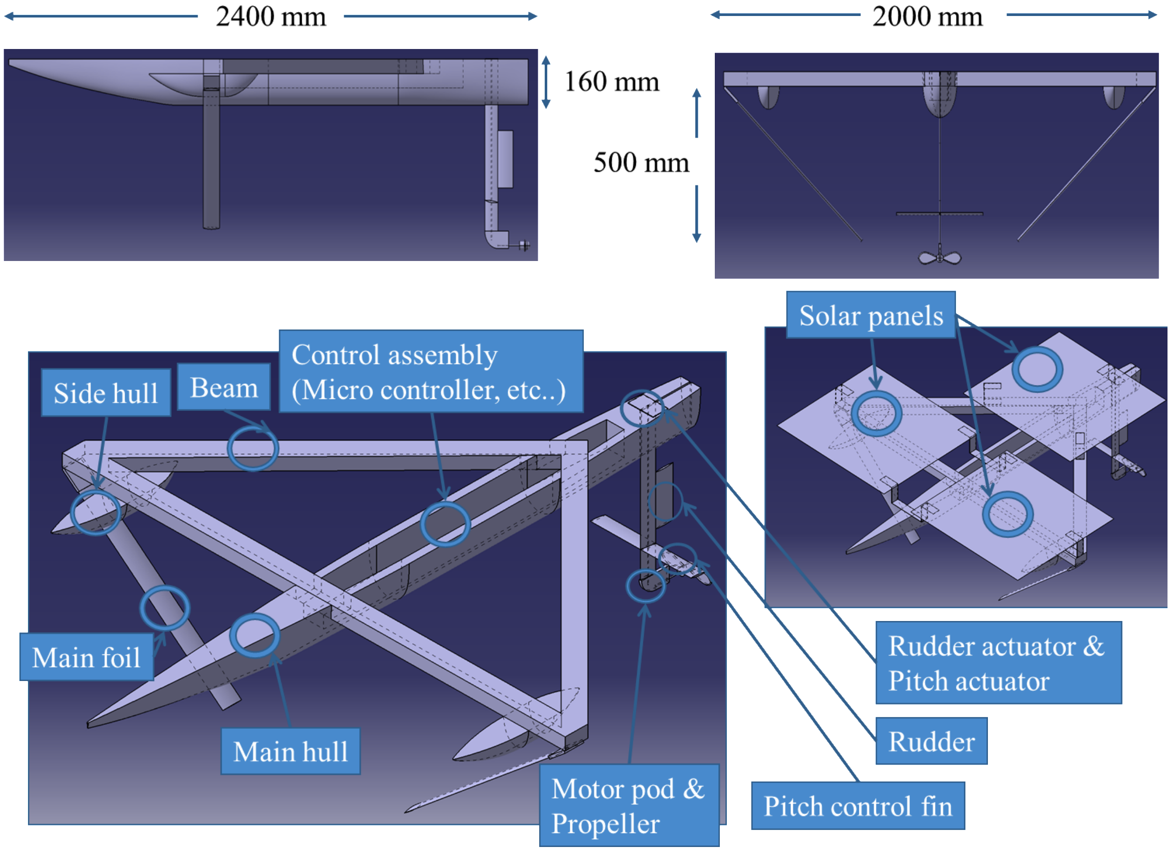

Figure 3: Overview 2014-A design

ANALYISING AND REDESINGING THE DEVELOPMENT PROCESS WITH THE V-MODEL

The V-Model

The V-Model (or “Vee Model” as a less common terminology) is well known and used within the Systems Engineering community. As per (INCOSE, 2011):

“The Vee model is used to visualize the system engineering focus, particularly during the Concept and Development Stages. The Vee highlights the need to define verification plans during requirements development, the need for continuous validation with the stakeholders, and the importance of continuous risk and opportunity assessment”. As shown in Figure 4 the V-Model graphically displays the decomposition of a design into smaller sub system designs and component designs which are ultimately manufactured or procured and integrated into sub systems and assembled into a final finished product. All the while Verification and Validation activities will occur at the multiple layers of the product decomposition (for completeness definitions of Verification and Validation are provided in Figure 5). While a single conceptual V (such as Figure 4) conveys significant level of information with regard to a Systems Engineering methodology of top down design followed by bottom up integration if a project team is to use it to convey the information related to their particular project (in planning, ongoing development or retrospective review) the diagram can quickly become cluttered resulting in it being difficult to understand.

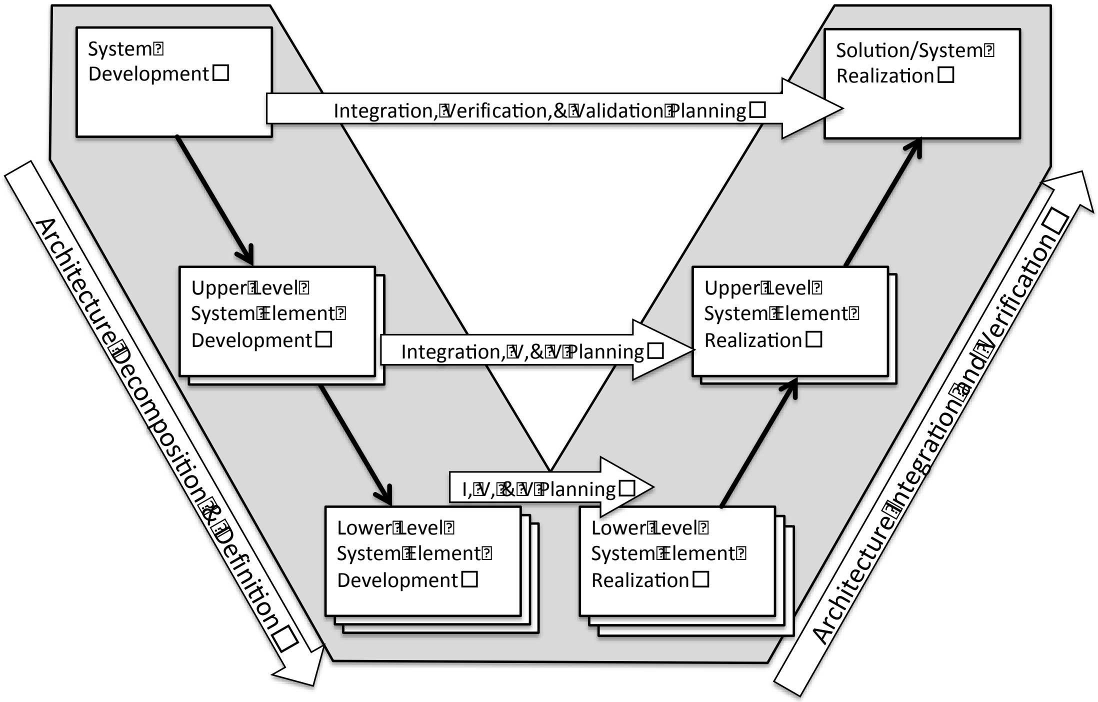
System Development
Integration, Verification, & Validation Planning
Solution/System Realization
Upper Level System Element Development
Integration, V, & V Planning
Upper Level System Element Realization
I, V, & V Planning
Lower Level System Element Development
Lower Level System Element Realization
Architecture Decomposition & Definition
Architecture Integration and Verification

Figure 4: A simplified V-Model Source: (INCOSE, 2011)

Verification of a product shows proof of compliance with requirements that the product can meet each “shall” statement as proven th[r]ough performance of a test, analysis, inspection, or demonstration.

Validation of a product shows that the product accomplishes the intended purpose in the intended environment that it meets the expectations of the customer and other stakeholders as shown through performance of a test, analysis, inspection, or demonstration.

Figure 5: Verification and Validation definitions Source: (NASA, 2007)

In (Scheithauer & Forsberg, 2013) a set of consistent V-Model views are proposed to allow project development teams to adequately describe their projects by way of the V-Model without adding unnecessary complexity to a single V. Their aim is to update the V-Model (from its roots in the 1980’s independently being invented by NASA and a collaboration of Hal Mooz and Kevin Forsberg) such that there is no longer a requirement for excessive customization and allow the V-Model to better cope with more modern approaches to systems development. The four views described by (Scheithauer & Forsberg, 2013) are described as: The Basic V (a standardized system of architectural decomposition), The Development V (hand-over of information within the system architecture), The Assurance V (Verification and Validation activities) and The Dynamic V (displays the iterations occurring over the product development).

In this paper the authors apply the V-Model views proposed by (Scheithauer & Forsberg, 2013) to the boat development project retrospectively, analyze the results and use this to generate an improved project development process. At time of writing it is not known to the authors of any publicly available examples of using the V-Model views proposed by (Scheithauer & Forsberg, 2013) so it is hoped this paper can act as to as an aid for implementers trying to make use of them.

Initial Analysis of 2014 Development with the Basic-V

The Basic-V as described in (Scheithauer & Forsberg, 2013) provides a standardized way of architectural decomposition. Applying to the solar boat project identified the project as one of product *and* service in that the boat must be built *and* raced. (Scheithauer & Forsberg, 2013) advocate the use of the Basic-V with left to right on the diagram being a logical sequence such that the "V" does not become distorted. However for the purposes of this research on the Basic-V the logical sequence is replaced with time as depicted in Figure 6. Indication of the type of activity performed is indicated by the color of the rectangle on Figure 6 such that the information about what side of the V is being depicted is not lost. Additional project management information is provided at the bottom of Figure 6. The process of displaying the project timeline with architectural decomposition offers a greater sense of understanding of the different aspects of the project than simpler project timelines (such as Figure 1) or Gantt charts. For example it can be seen that the bulk of the work on the lowest level of the hierarchy occurred not long before the race. However when such a representation is used by a systems engineer attempting to build a new product development process it is not clear what changes are needed.

In (Scheithauer, 2012) a Knowledge Growth Curve is presented as a way of visualizing the amount of knowledge the team has about the system under development. Figure 7, although highly subjective is an attempt to visualize this for the 2014 boat development where team skill has been incorporated with knowledge, in an attempt to quantify the uncertainty in the project. Clearly it should be noted that the rate of learning increased significantly to the end of the project with the first three months not represent particularly fast learning.

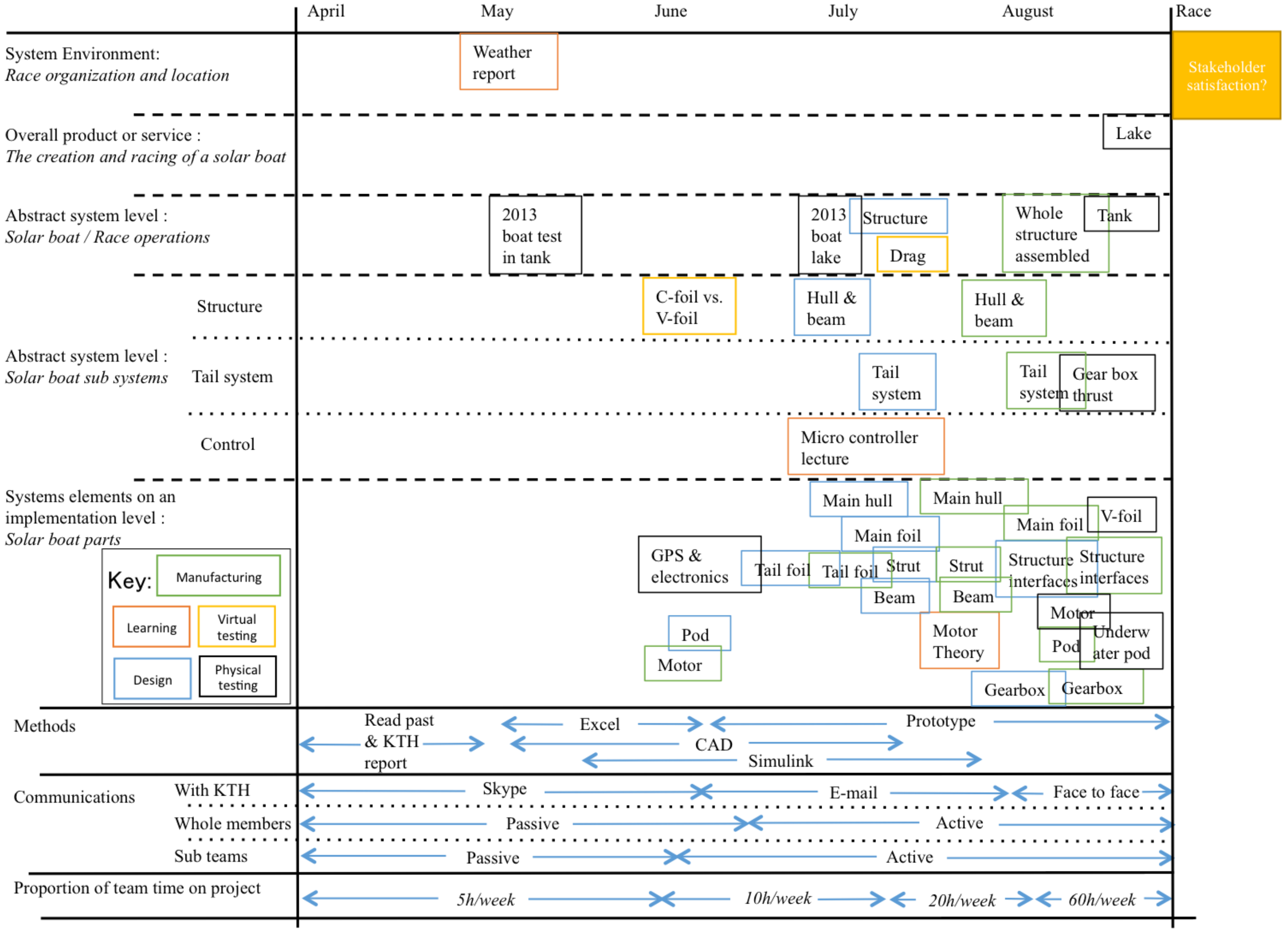

Figure 6: Basic-V decomposition and project activities for 2014 development

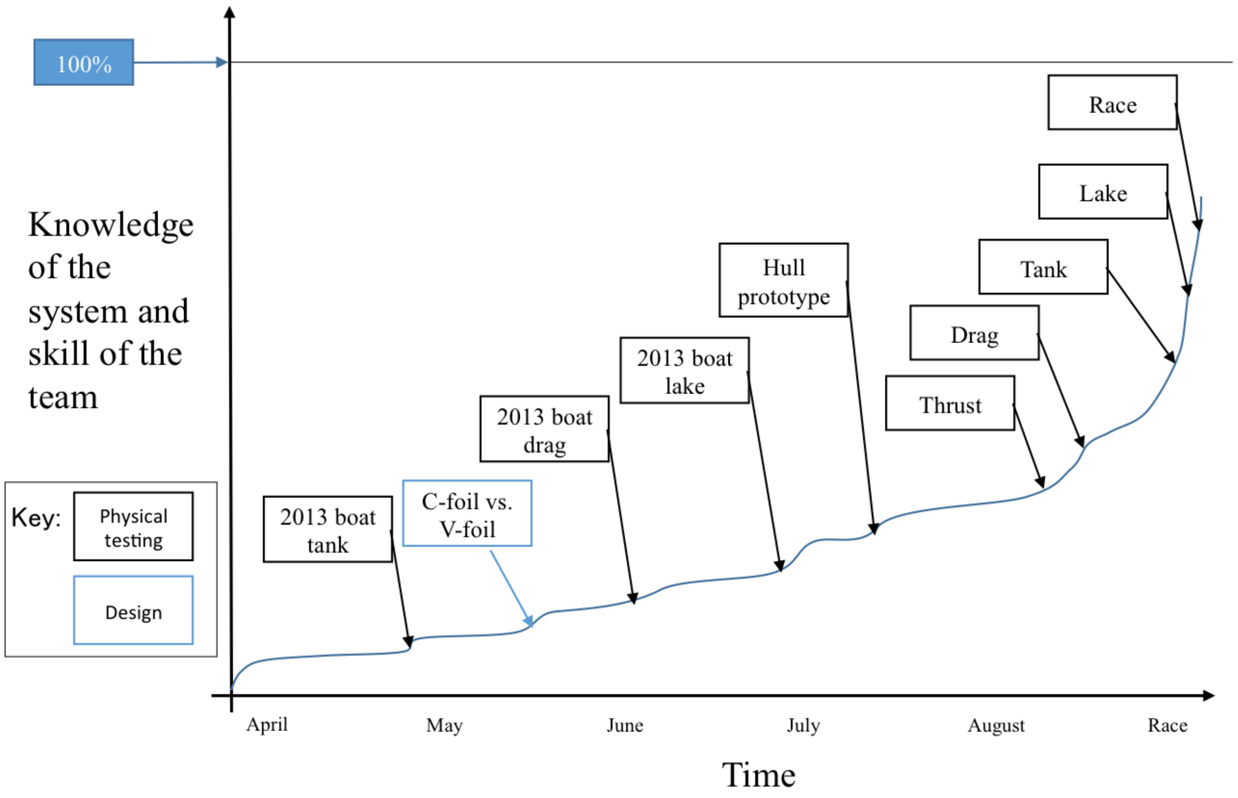

Figure 7: Knowledge / Skill growth for 2014 development

Reviewing and Improving Activity on the Dynamic-V

The Dynamic-V as described in (Scheithauer & Forsberg, 2013) displays the iterations occurring over a product development. Figure 8 displays this for 2014 boat development. It is noted that all the iterations which were performed were event driven rather than proactively planned. The red loop which spans the entire system hierarchy represents the attempt to correct a major yaw control fault which was only discovered during a lake test one day before the race, and was still present despite modifications after the first day of racing. The lower level red loop represents the integration of a gearbox which was not included in the original design but was required to obtain sufficient thrust from the selected motor. Reviewing the smaller

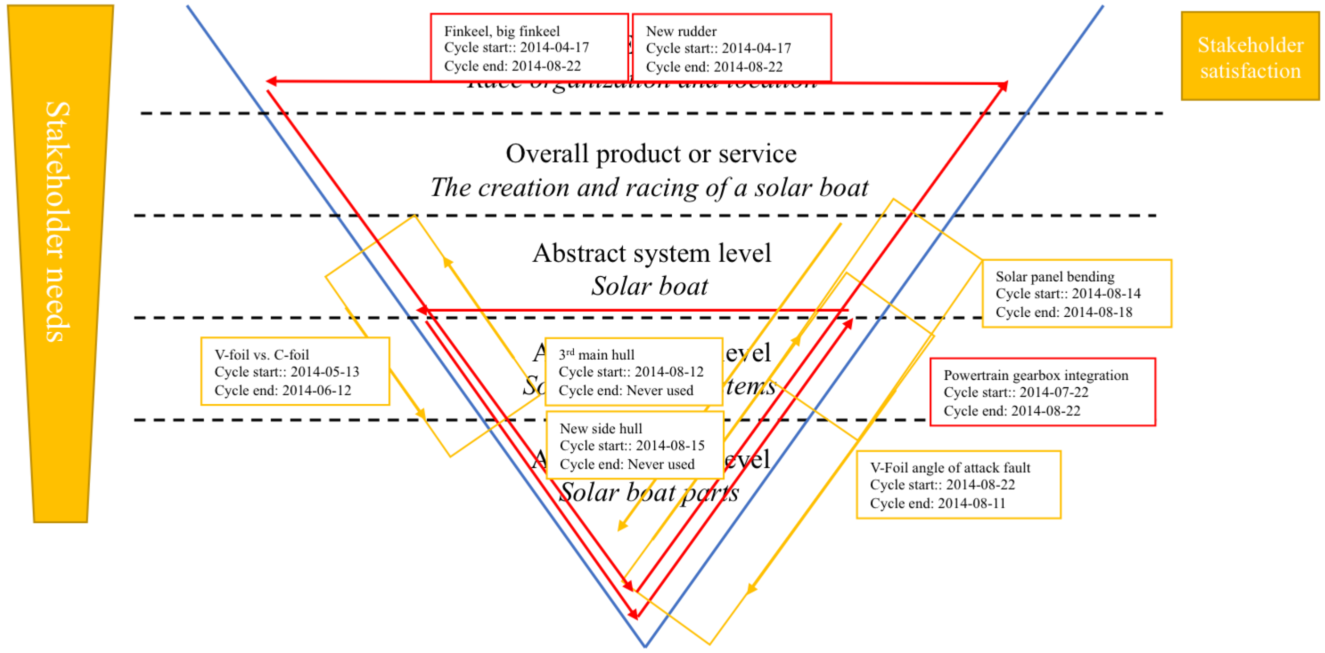

Figure 8: Dynamic-V for 2014 development

loops; on the design side different foil configuration concepts were simulated while on the systems integration side incompatibilities were found with parts resulting in modifications.

The Dynamic-V presented for the 2014 project (Figure 8) is rebuilt as (Figure 9) with two proactively defined loops. Here explicitly a loop is defined to build a prototype prior to the race. As described in (Scheithauer, 2012) "iterations are the rule in systems engineering, not the exceptions." Given the design had multiple faults which were not discovered until the race an explicit prototype loop should be included. The design side iteration is replaced with a more general one, to indicate the need to develop designs for all sub systems at least at a concept level and iterate the overall design prior to manufacturing.

Reviewing and Improving Activity on the Development-V

The Development-V as described in (Scheithauer & Forsberg, 2013) describes hand-over of information within the system

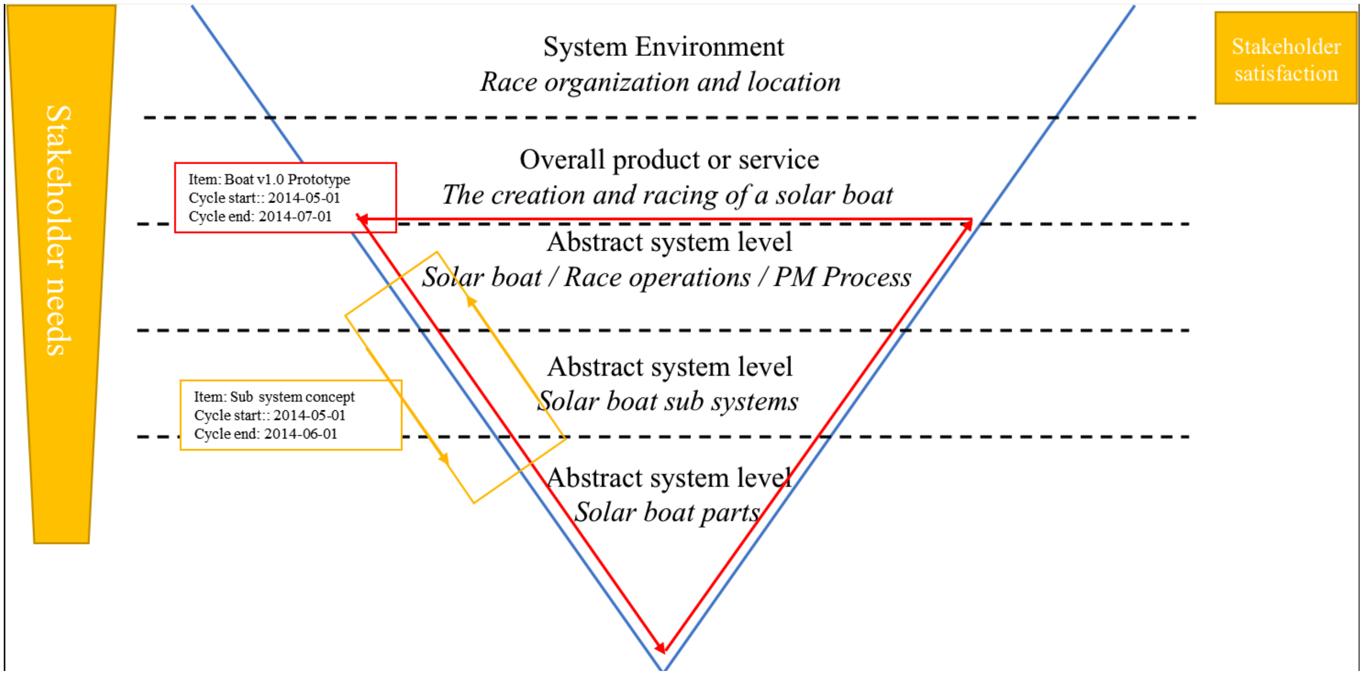

Figure 9: Dynamic-V for an improved process

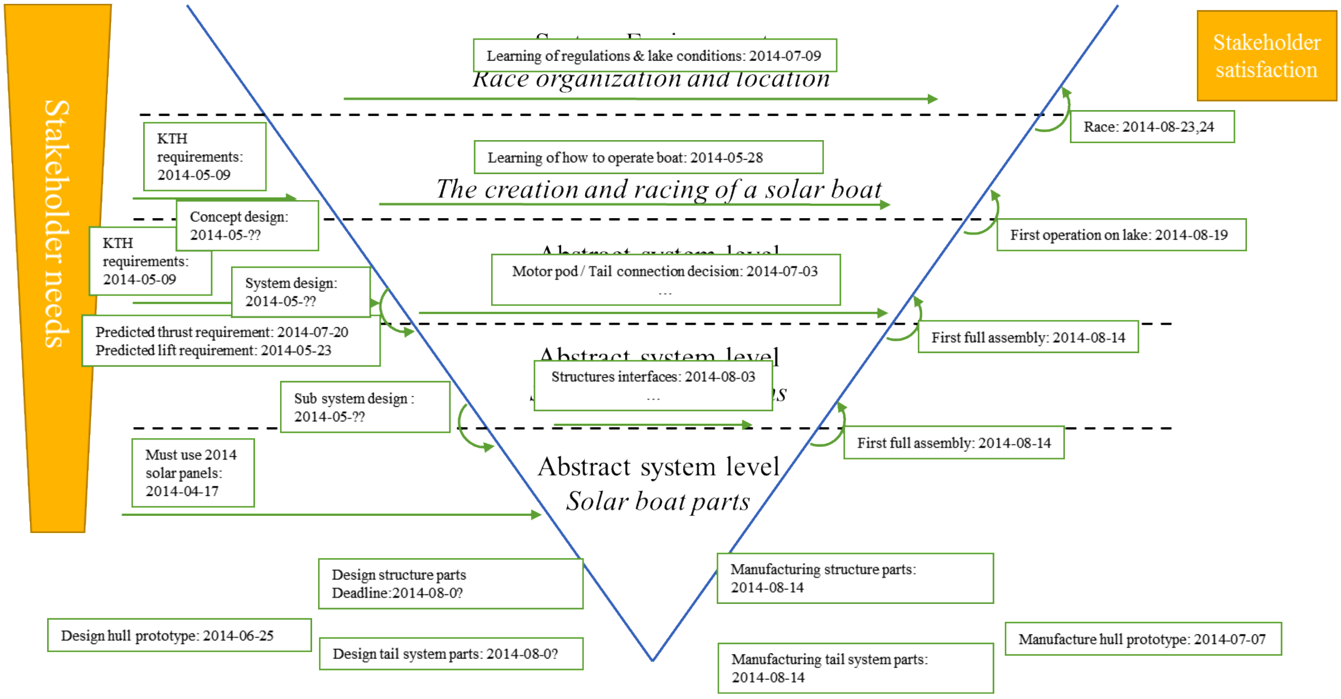
Stakeholder needs
Learning of regulations & lake conditions: 2014-07-09
Race organization and location
Stakeholder satisfaction
Race: 2014-08-23,24
KTH requirements: 2014-05-09
Learning of how to operate boat: 2014-05-28
The creation and racing of a solar boat
Concept design: 2014-05-??
First operation on lake: 2014-08-19
KTH requirements: 2014-05-09
System design: 2014-05-??
Motor pod / Tail connection decision: 2014-07-03
…
Predicted thrust requirement: 2014-07-20
Predicted lift requirement: 2014-05-23
First full assembly: 2014-08-14
Abstract system level
Structures interfaces: 2014-08-03
…
Sub system design : 2014-05-??
First full assembly: 2014-08-14
Must use 2014 solar panels: 2014-04-17
Abstract system level
Solar boat parts
Design structure parts
Deadline:2014-08-0?
Manufacturing structure parts: 2014-08-14
Design hull prototype: 2014-06-25
Design tail system parts: 2014-08-0?
Manufacture hull prototype: 2014-07-07
Manufacturing tail system parts: 2014-08-14

Figure 10: Development-V for 2014 development

architecture and Figure 10 displays this for the 2014 boat development. The following points of interest are observed: Very few requirements filtered down through the left side of the V, multiple design work products such as “System design” did not have completion dates known to anyone on the team (marked as ?) and systems integration occurred rapidly from bottom of the V to the top not long before the race.

The Development-V for 2014 is modified and displayed in Figure 11. Multiple additional work products have been defined on the design side and implementation side, completion dates given to work products and requirements now flow down the design side.

Reviewing and Improving Activity on the Assurance-V

The Assurance-V as described in (Scheithauer & Forsberg, 2013) displays verification and validation activities, Figure

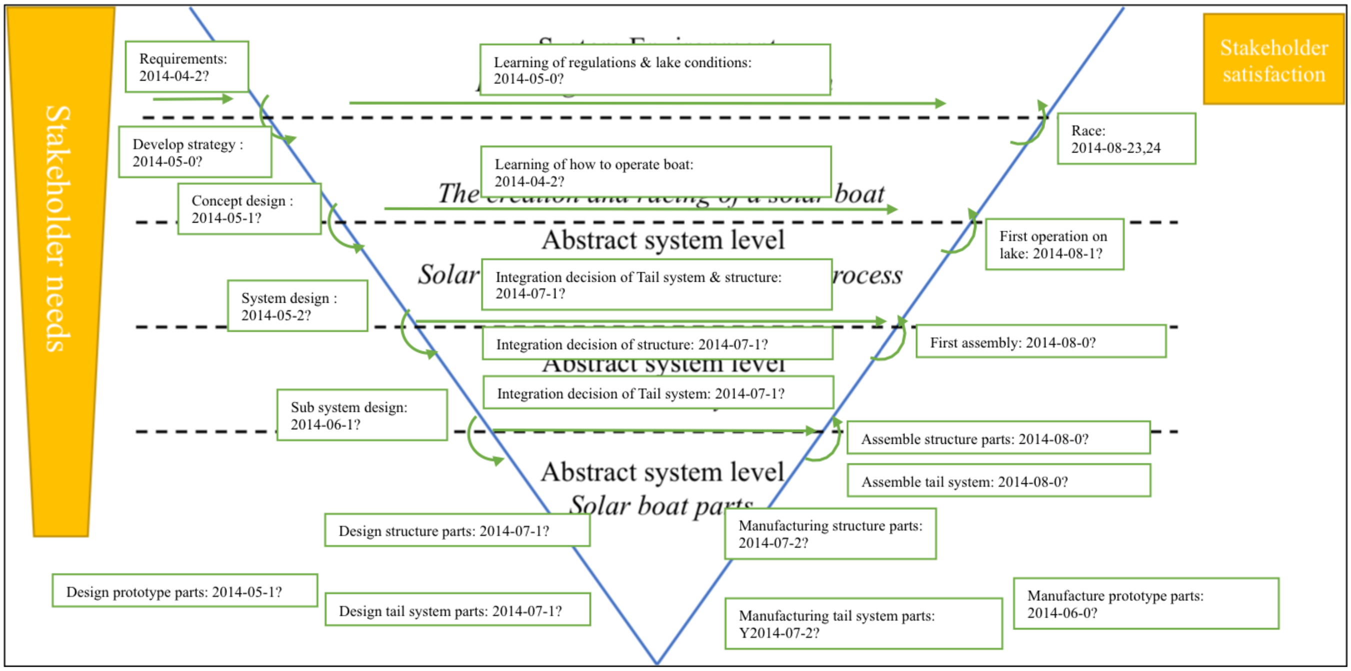
Stakeholder needs
Requirements: 2014-04-2?
Learning of regulations & lake conditions: 2014-05-0?
Stakeholder satisfaction
Develop strategy : 2014-05-0?
Race: 2014-08-23,24
Learning of how to operate boat: 2014-04-2?
The
boat
Concept design : 2014-05-1?
Abstract system level
First operation on lake: 2014-08-1?
Solar
Integration decision of Tail system & structure: 2014-07-1?
rocess
System design : 2014-05-2?
Integration decision of structure: 2014-07-1?
First assembly: 2014-08-0?
Integration decision of Tail system: 2014-07-1?
Sub system design: 2014-06-1?
Assemble structure parts: 2014-08-0?
Abstract system level
Assemble tail system: 2014-08-0?
Design structure parts: 2014-07-1?
Manufacturing structure parts: 2014-07-2?
Design prototype parts: 2014-05-1?
Design tail system parts: 2014-07-1?
Manufacturing tail system parts: Y2014-07-2?
Manufacture prototype parts: 2014-06-0?

Figure 11: Development-V for an improved process

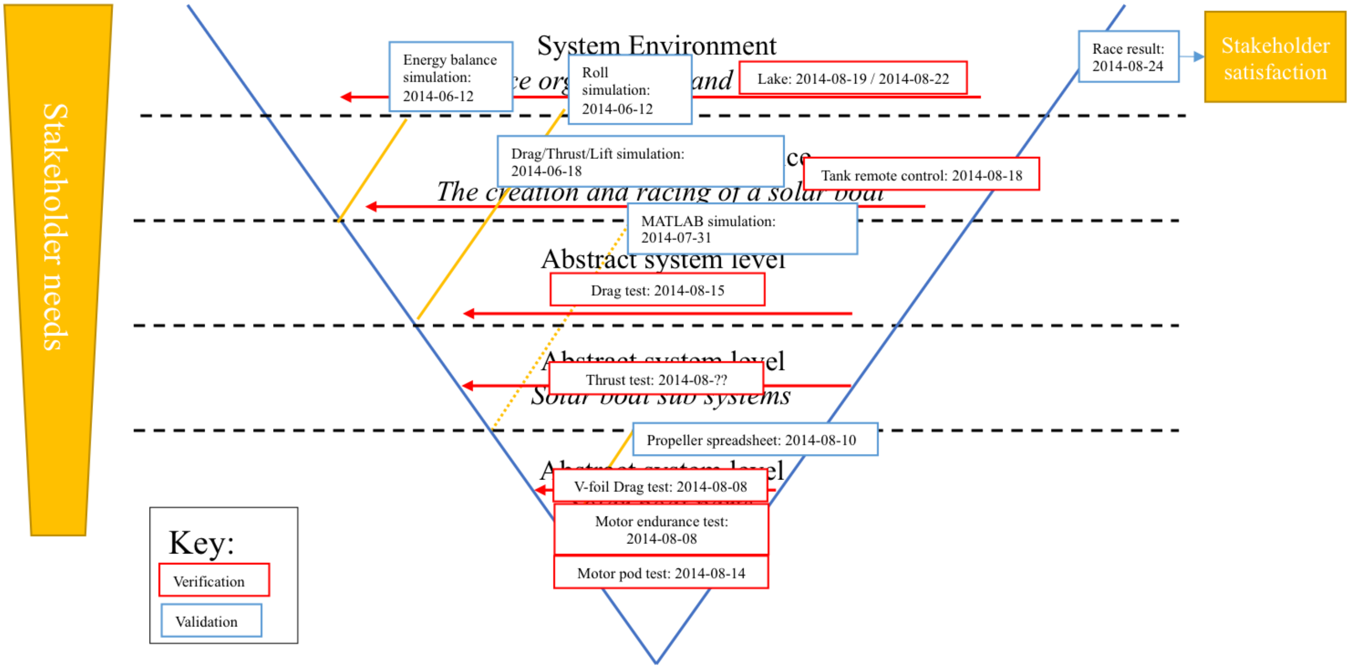
Stakeholder needs
System Environment
Energy balance simulation: 2014-06-12
Roll simulation: 2014-06-12
and
Lake: 2014-08-19 / 2014-08-22
Race result: 2014-08-24
Stakeholder satisfaction
Drag/Thrust/Lift simulation: 2014-06-18
Tank remote control: 2014-08-18
MATLAB simulation: 2014-07-31
Drag test: 2014-08-15
Thrust test: 2014-08-??
Propeller spreadsheet: 2014-08-10
V-foil Drag test: 2014-08-08
Motor endurance test: 2014-08-08
Motor pod test: 2014-08-14
Key:
Verification
Validation

Figure 12: Assurance-V for 2014 development

12 displays this for 2014 boat development. It can be noted that much of the verification occurred only shortly before the race itself and little to no validation work was performed on several critical systems.

The Assurance-V is rebuilt as multiple diagrams for clarity with each depicting a period of time to avoid any one diagram becoming too cluttered. These diagrams are shown in Figure 13 (April/May), Figure 14 (June), Figure 15 (July) and Figure 16 (August). Figure 13 (April/May) remains similar to what was conducted in 2014, however it should be noted that this validation work now links to work products displayed in the updated Development-V (Figure 11) such the work products can be marked as completed based on the success of the validation work.

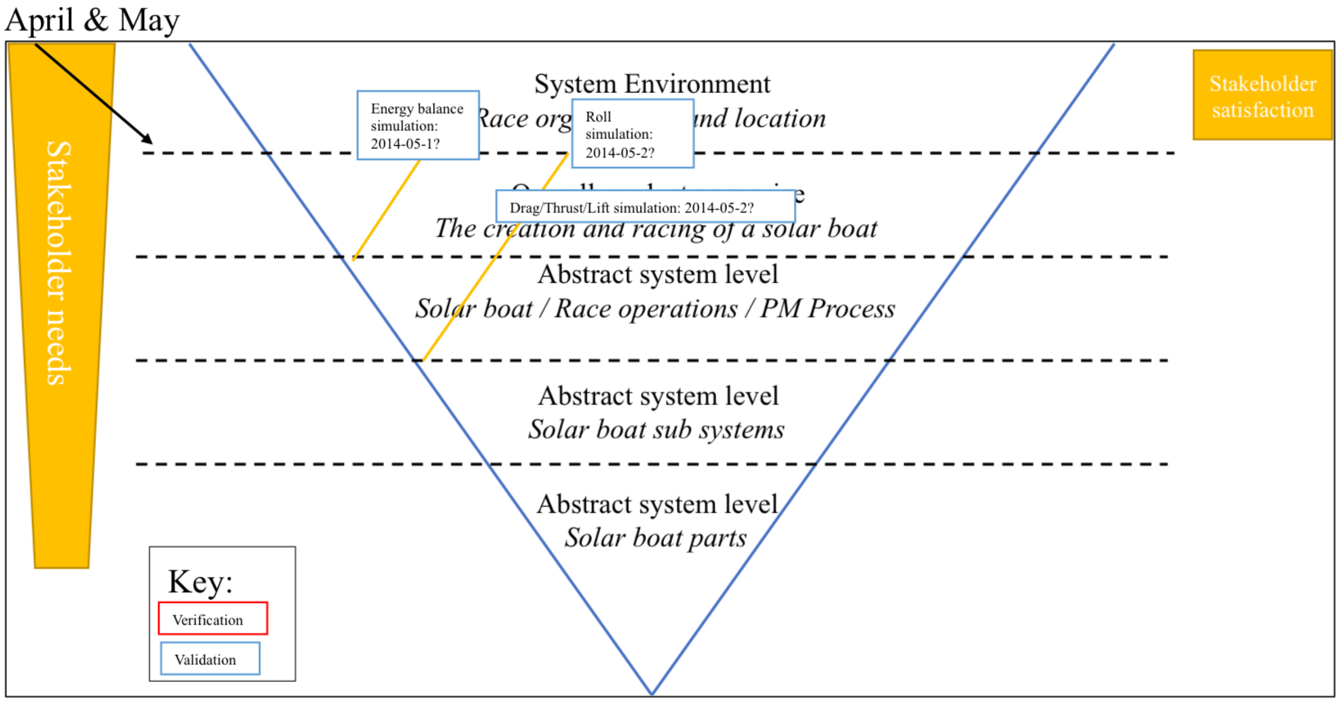
April & May
Stakeholder needs
Stakeholder satisfaction
System Environment
Race org... ...and location
Energy balance simulation: 2014-05-1?
Roll simulation: 2014-05-2?
Drag/Thrust/Lift simulation: 2014-05-2?
The creation and racing of a solar boat
Abstract system level
Solar boat / Race operations / PM Process
Abstract system level
Solar boat sub systems
Abstract system level
Solar boat parts
Key:
Verification
Validation

Figure 13: Assurance-V for an improved process (April/May)

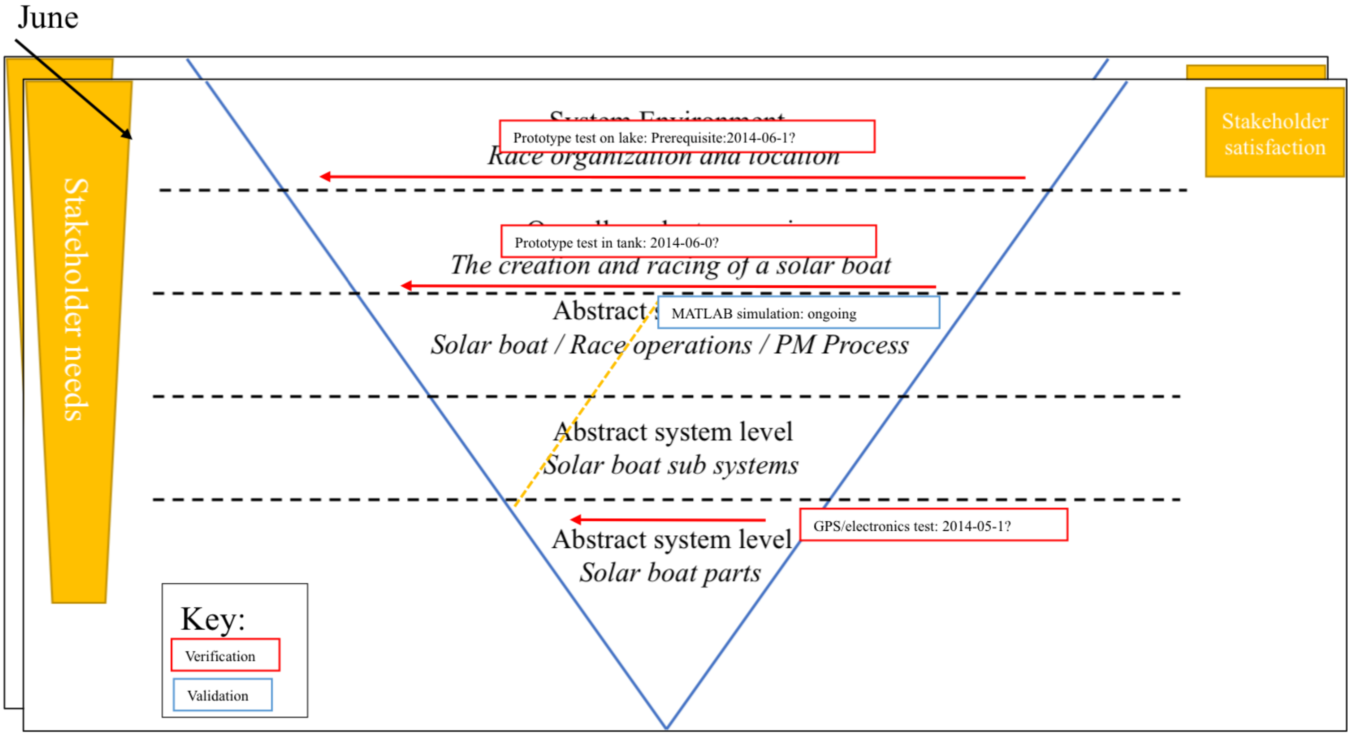
June
Stakeholder needs
Stakeholder satisfaction
Prototype test on lake: Prerequisite:2014-06-1?
Race organization and location
Prototype test in tank: 2014-06-0?
The creation and racing of a solar boat
Abstract
MATLAB simulation: ongoing
Solar boat / Race operations / PM Process
Abstract system level
Solar boat sub systems
GPS/electronics test: 2014-05-1?
Abstract system level
Solar boat parts
Key:
Verification
Validation

Figure 14: Assurance-V for an improved process (June)

Figure 14 (June) has the addition of verification testing of the prototype and basic validation of the boat design by simulation of the boat.

Figure 15 (July) depicts a significant increase in verification testing which occurs as parts are manufactured and integrated

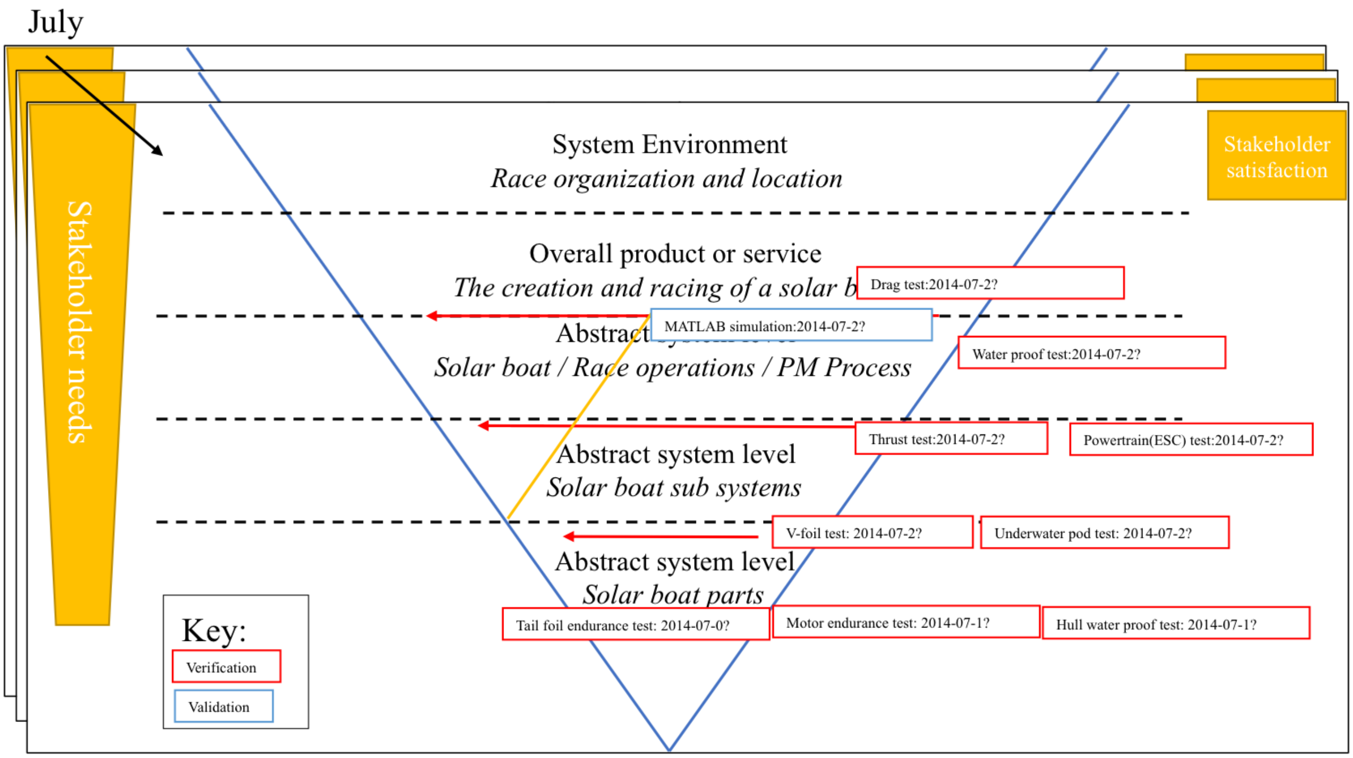
July
Stakeholder needs
System Environment
Race organization and location
Stakeholder satisfaction
Overall product or service
The creation and racing of a solar b
Drag test:2014-07-2?
MATLAB simulation:2014-07-2?
Abstract
Solar boat / Race operations / PM Process
Water proof test:2014-07-2?
Thrust test:2014-07-2?
Powertrain(ESC) test:2014-07-2?
Abstract system level
Solar boat sub systems
V-foil test: 2014-07-2?
Underwater pod test: 2014-07-2?
Abstract system level
Solar boat parts
Tail foil endurance test: 2014-07-0?
Motor endurance test: 2014-07-1?
Hull water proof test: 2014-07-1?
Key:
Verification
Validation

Figure 15: Assurance-V for an improved process (July)

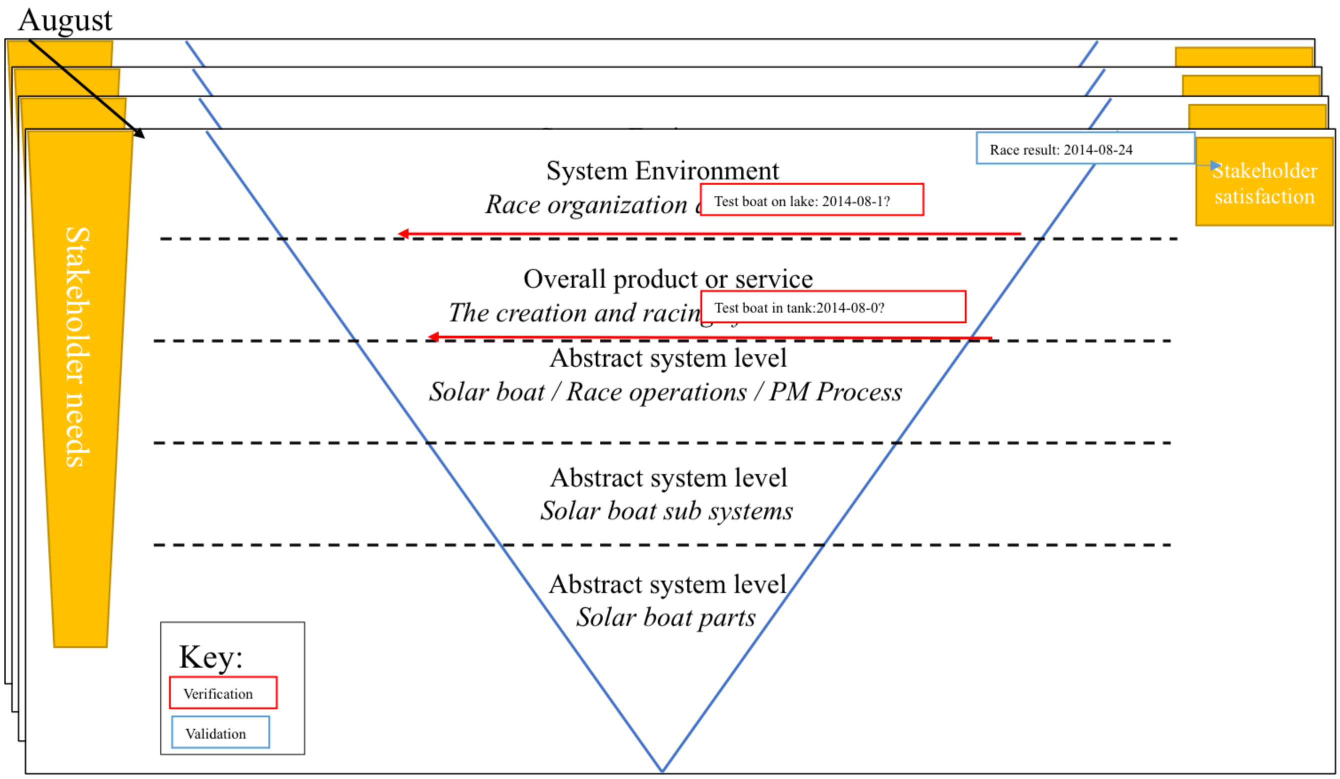

Figure 16: Assurance-V for improved process (August)

into sub systems. The validation simulation is improved based on data gathering during the verification testing.

Figure 16 (August) depicts the verification testing of the boat prior to the race.

Rebuilding the Project Plan on the Basic-V

Based on the activity changes derived from the multiple V-Model changes on the views described previously, the Basic-V view for recording the project activity in Figure 6 is rebuilt as Figure 17. This work schedule is considered to be a more productive product development process as Systems Engineering activities missing or occurring at inappropriate times are rectified.

LOOKING BEYOND THE V-MODEL FOR DEVELOPMENT IMPROVEMENTS

The four V-Model views have been used in this paper to explain some failings of the development process and how to potentially improve the process for future development projects. However, product development and Systems Engineering are very diverse disciplines and therefore no single model or diagram likely can be used to explain and correct everything in a development project. In this section other factors seen by industry as important for effective systems development are explored and compared to the development conducted by the team, proposed improvements to the methods used by the team are then provided.

In (Oppenheim et al., 2011) by way of presenting data from various US space and defense development programs Systems Engineering is presented as not always being delivered effectively, with a somewhat shocking statistic presented in that on government product development programs charged time is frequently approximately 60-90% waste. To enable more

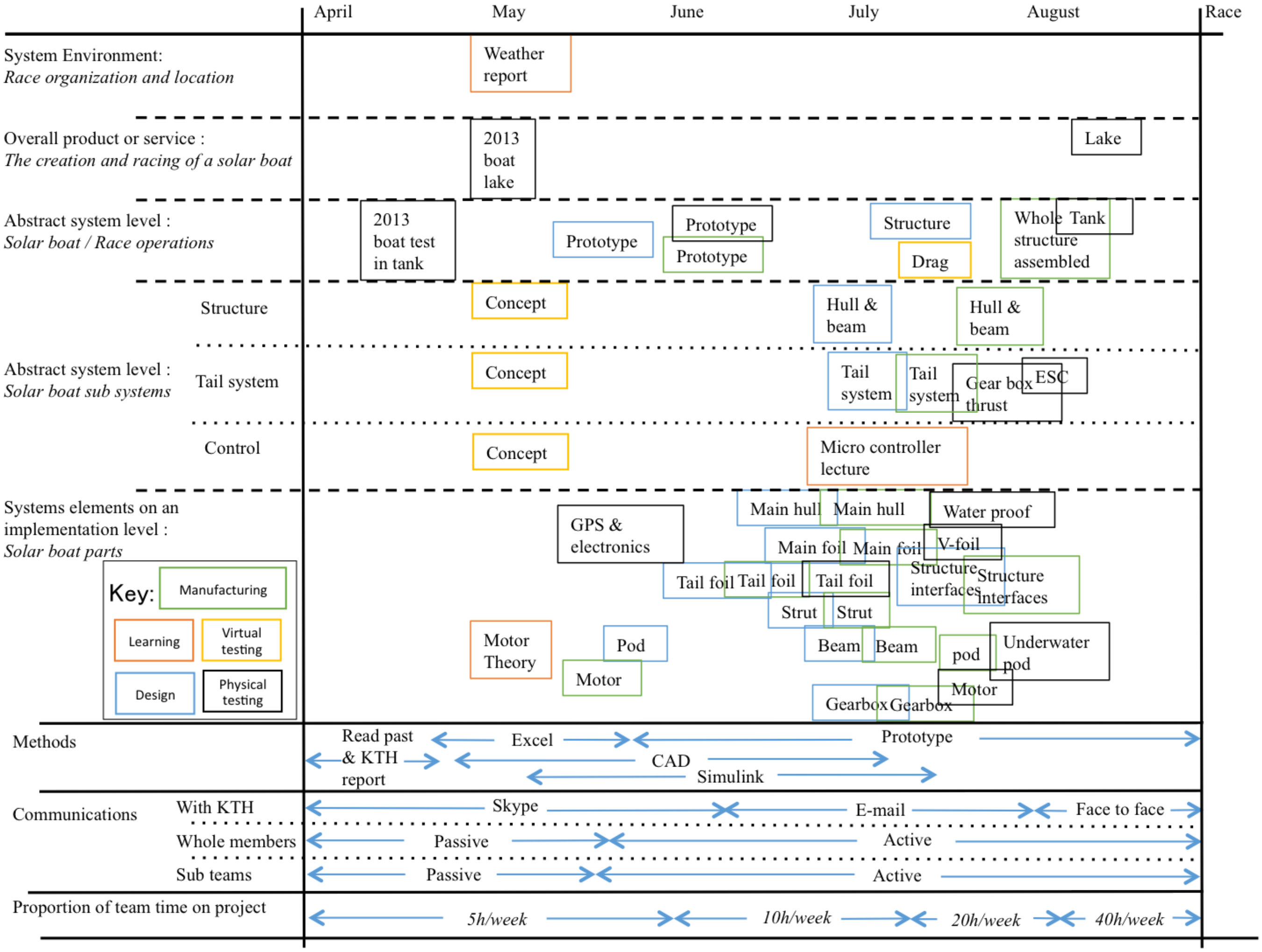

Figure 17: Basic-V decomposition and project activities for an improved process

effective Systems Engineering multiple *Lean Enablers* are proposed by (Oppenheim et al., 2011) as a comprehensive checklist suggested to enable a product development program to take a *Lean Systems Engineering* approach. *Lean Thinking* is defined in (INCOSE, 2011) to be "the dynamic, knowledge-driven, and customer focused process through which all people in a defined enterprise continuously eliminate waste with the goal of creating value.". With Lean Systems Engineering being defined as "the application of Lean principles, practices, and tools to SE and to the related aspects of enterprise management (EM) in order to enhance the delivery of value (which is defined as flawless delivery of product or mission with satisfaction of all stakeholders) while reducing waste". The checklist developed by (Oppenheim et al., 2011) is comprehensive and long, but the enablers are mapped to an existing set of NASA Key Enablers for Successful Programs in Aerospace and NASA Best Practices of Top Performing Aerospace Companies. As a method of providing insight into potential improvements to the development process these NASA findings are compared to the 2014 development to find proposed changes. Further the Lean Enablers thought to provide greatest benefit to the solar boat development are selected and presented.

NASA Key Enablers for Successful Programs in Aerospace Compared to the 2014 Development Process

Table 1 compares the NASA Key Enablers for Successful Programs in Aerospace to the experiences on the solar boat development project. Broadly effectiveness at delivering these was low, with Capability Maturity and Design Robustness Mindset being seen potentially most important for improving performance. As described in the section on V-Model decomposition greater use of prototypes is likely to improve this situation.

NASA Best Practices of Top Performing Aerospace Companies Compared to the 2014 Development Process

In Table 2 the NASA Best Practices of Top Performing Aerospace Companies are compared to the experiences on the solar boat development project. Again effectiveness in 2014 is seen as low. Operational points of note include the use of models and simulation, development of metrics to measure the system and the building of a continuous improvement culture.

Table 1: NASA Key Enablers for Successful Programs in Aerospace compared to the project process

NASA "Key Enablers of Successful Programs" (Oppenheim et al., 2011)	**Effectiveness in 2014**	**Proposed changes**
Visionary Leadership: Role of organizational leadership in establishing a clear overarching purpose, deriving and articulating a compelling but credible vision to fulfill that purpose.	OK. Clear goal to win the race, but priority of winning race vs. developing a very advanced boat was not clear.	Greater use of prototypes to ensure basic designs are ready for racing before developing more advanced designs.
Capability Maturity: Organization attainment of high levels of "Capability Maturity" to support and facilitate the undertaking of complex systems development.	Low. As most work was completed only once.	Greater use of prototypes to drive skills up across the V earlier into the project. Make more formal use of past year team members experience. Set individual team members the task of honing a particular skill set.
Systems Engineering Culture: A pervasive mental state and bias for Systems Engineering methods applied to problem solving across the development lifecycle and at all levels of enterprise processes.	Low. Most work was conducted by sub teams in isolation.	Use of the V-Model to define the work product requirements with the team reviewing progress as one group regularly.
Design Robustness Mindset: High levels of focus on system safety and reliability driven by a bias toward achieving robustness, supported by the cultural attitude of "Failure is not an Option".	Low. Multiple failures occurred during the race.	Iterative prototypes to start verification testing early.
Accountability Structure: Effective decision making accomplished through clearly defined structures of assigned responsibility and accountability for decisions at appropriate levels and phases of system development.	Low. It was not clear when work products were complete and who was responsible to maintain them.	Important work products defined on the Development-V with individuals allocated to ensuring they are delivered.

A Selection of Lean Enablers Viewed as Important to the Improvement of Solar Boat Development

A selection of Lean Enablers seen as offering greatest improvement to solar boat development from (Oppenheim et al., 2011) are presented with commentary in this section.

Value: Lean Enablers seen as important to the solar boat development are described in Table 3 it is possible to see that the project had considerable confusion over what value was as there was a bias to attempt to build a very high performance boat to maximize potential value but which ultimately sacrificed robustness of the craft when in the race.

Map the Value Stream: Lean Enablers considered important are listed in Table 4. Value Stream mapping was not conducted in the 2014 development, instead a fixed Gantt chart was used to plan task allocation. The only metric of progress was adherence to this plan. A major benefit is expected, if currently delivered value is monitored and used to drive team decision making, based on what the team is measured to be capable of.

Flow: Lean Enablers are detailed in Table 3 and provide pointers on how to operate during product development. With de-

Table 2: NASA Best Practices of Top Performing Aerospace Companies

NASA "Best Practices" (Oppenheim et al., 2011)	**Effectiveness in 2014**	**Proposed changes**
Leading with Vision: Sharing the Vision, Providing Goals, Direction & Visible Commitment	OK. Clear goal to win the race, but priority of winning race vs. developing a very advanced boat was not so clear.	Greater use of prototypes to ensure basic designs are ready for racing before developing more advanced designs.
Focusing on Requirements: Mission Success Driven Requirements & Validation Process	Low. Original requirements document was not attainable, read once and not looked at again.	Set a few attainable requirements at each level of the architecture decomposition and ensure they are verified and validated.
Achieving Robust Systems: By Rigorous Analysis, Robustness of Design, HALT/HASS testing	Low. Multiple failures occurred during the race.	Iterative prototypes to start verification testing early.
Models & Simulation: Model-based Systems Engineering with "seamless" models, validated with Experts	OK. Models were developed for some systems, but development was slow and not for all systems. Integration was not seamless.	Each system to have a minimum level of modeling.
Visible Metrics: Effective measures, visible supporting data for better decisions at each organizational level	Low. Generally based on deviation from fixed plan.	Develop meaningful metrics to quantify team progress.
Systems Management: Managing for Value & Excellence throughout the Life-cycle	Low. Validation and verification stages were compressed.	By way of prototypes develop across the whole V from the beginning.
Building Culture: Based on Foundation "Systems" Principles, Continuous improvement	Low. Generally work was completed in silos.	Ensure team understands that little knowledge is known at the start of the project and it is our job to increase it. Develop a continuous improvement program to improve the development process as we work.

Table 3: Enablers for Lean Principle 1: Value

Enablers source: (Oppenheim et al., 2011)	**Effectiveness in 2014**	**Proposed changes**
1.2 Establish the Value of the End Product or System to the Customer	Multiple definitions of value were being worked towards, including to win the race and to maximize potential performance to increase student learning.	At the project startup survey the students to learn their interests and use this to define value.

risking early, collaborating effectively & efficiently and monitoring progress considered to be important.

Pull: Lean Enablers listed in Table 6 indicate the importance of only completing work which is needed by a downstream system which can avoid the creation of technology which is not needed.
Perfection: Lean Enablers detailed in Table 7 detail the need to implement continuous improvement by utilizing the talent of all the team, make good use of past programs and enable standardization of processes. For 2014 development no continuous improvement system was implemented.

Perfection: Lean Enablers listed in Table 8 emphasizing the need for learning. Arguably the most important for a student project as this one of the measures of value itself. As described by (Scheithauer, 2012) increasing the knowledge of the system is important for its successful delivery. As the approximation to the Knowledge Growth Curve in Figure 7 shows much of the learning about the system occurred towards the end of the project as the systems were integrated and went through verification testing.

Table 4: Enablers for Lean Principle 2: Map the Value Stream (Plan the Program)

Enablers source: (Oppenheim et al., 2011)	**Effectiveness in 2014**	**Proposed changes**
2.2 Map the SE and PD Value Streams and Eliminate Non-Value Added Elements	Value stream was not created.	Develop a value stream for each project iteration.
2.3 Plan for Frontloading the Program	Planned to frontload learning but task allocation for this was poor.	Frontload tasks across all aspects of the V-Model to maximize learning. E.g. testing of past years boat. Recursively develop a prototype so system verification activity starts early.
2.4 Plan to Develop Only What Needs Developing	Technology was developed which was not needed. E.g. height sensor.	Identify the need before developing a technology.
2.6 Plan Leading Indicators and Metrics to Manage the Program	Not followed. Project manager found it difficult to understand project progress, especially as team is not co-located.	Develop meaningful metrics which are updated and reviewed weekly. E.g. time on project, wasted time and requirements met

Table 5: Enablers for Lean Principle 3: Flow

Enablers source: (Oppenheim et al., 2011)	**Effectiveness in 2014**	**Proposed changes**
3.2 Clarify, Derive, Prioritize Requirements Early and Often During Execution	A requirements document was created by KTH students, but it was more of a “wish list” than a rigorously analyzed and generated document.	Set a few attainable requirements at each level of the architecture decomposition.
3.3 Front Load Architectural Design and Implementation	Structural architecture inherited from KTH was sound. Powertrain architecture was not set early. Validation of design was patchy.	Define the architecture, validate it numerically and by small experiment. De-risk new technologies with minimum viable implementations.
3.4 Systems Engineers to accept Responsibility for coordination of PD Activities	Coordination of project development was done by the project manager.	Use work products on the Development-V and their dependencies on one another to define the coordination activities.
3.5 Use Efficient and Effective Communication and Coordination	One large passive weekly meeting. Document sharing. Video conference and Skype with KTH. Japanese / English language barrier was large and problematic.	One large active weekly meeting where all members share activity reports. Defined key work products and ensure they are always available for consultation. Share key metrics. Actively recruit a team member to work as a translator.
3.7 Make Program Progress Visible to All	Progress displayed as adherence to a fixed plan which was displayed infrequently.	Use clear metrics on posters indicating progress towards development goals.

CONCLUSIONS

In this paper, an overview of the development activities which resulted in the creation and racing of a solar powered autonomous boat as part of a student design project have been presented. Retrospective analysis of the development activities was conducted by mapping completed activity to:

- The four V-Model views presented in (Scheithauer & Forsberg, 2013)
- Best practices documented by NASA and presented by (Oppenheim et al., 2011)
- The Lean Systems Engineering enablers defined by (Oppenheim et al., 2011)

Based on this retrospective analysis an alternative development process is presented which is considered to likely increase the delivered value by the project.

The authors found using the V-Model views presented by (Scheithauer & Forsberg, 2013) to be an effective way to present development compared to using a single V-Model and hope these views can become standardized across industry. The Basic-V presented (Scheithauer & Forsberg, 2013) was used to decompose the system components, but project activity was tracked

Table 6: Enablers for Lean Principle 4: Pull

Enablers source: (Oppenheim et al., 2011)	**Effectiveness in 2014**	**Proposed changes**
4.2 Pull Tasks and Outputs Based on Need, and Reject Others as Waste	Not followed. Work was largely pushed.	Based on prototypes pull through development activities to deliver value.

Table 7: Enablers for Lean Principle 5: Perfection

Enablers source: (Oppenheim et al., 2011)	**Effectiveness in 2014**	**Proposed changes**
5.1 Pursue Continuous Improvement according to the INCOSE Handbook Process.	Not followed. No attempt to develop continuous improvement.	Develop a continuous improvement plan, referencing INCOSE Handbook for advice.
5.3 Use Lessons Learned from Past Programs for Future Programs	Architecture lessons learned adopted but not process lessons learned.	Build better development programs based on the findings of this paper. Consult with past team members more formally.
5.5 For Every Program Use a Chief Engineer Role to Lead and Integrate the Program from Start to Finish	Such a role only defined mid-way in the project. Firefighting lead to the role not being fulfilled.	Appoint a Chief Engineer from the start of the project.
5.6 Drive out Waste through Design Standardization, Process Standardization, and Skill-Set Standardization	Some students became experts at certain items, but others balanced many different skills.	Appoint team members to become team expert at a particular skill, and to become the team point of contact for that skill.
5.7 Promote All Three Complementary Continuous Improvement Methods to Draw Best Energy and Creativity from All Employees: 5.7.1 Utilize and reward bottom up suggestions for solving employee-level problems; 5.7.2 Use quick response small Kaizen teams comprised of problem stakeholders for local problems and development of standards; 5.7.3 Use the formal large Six Sigma teams for the problems which cannot be addressed by the bottom-up and Kaizen improvement systems, and do not let the Six Sigma program destroy those systems.	5.7.1 Not followed 5.7.2 Not followed 5.7.3 Not followed	5.7.1 At weekly meeting create a forum for problem and solution sharing 5.7.2 Items which cannot be solved at weekly meeting to be distributed to a Kaizen team for improvement 5.7.3 Likely not applicable to this sort of project

Table 8: Enablers for Lean Principle 6: Respect for People

Enablers source: (Oppenheim et al., 2011)	**Effectiveness in 2014**	**Proposed changes**
6.3 Expect and Support Engineers to Strive for Technical Excellence	Multiple skills were not developed until late into the project. Test plans and data recording was unacceptably poor at times.	Define areas of reasonability and hold people to account for delivery. Survey team members to understand their skills and aspirations.
6.4 Nurture a Learning Environment	Student project so naturally a learning environment, however focus was on individuals doing analysis alone rather than learning how to integrate a real system.	As shown in the Knowledge Growth Curve (Figure 7) learning is very important for the project. When deficiencies in knowledge are identified plug them quickly.

by time running from left to right as is more typical in V-Model representations such as described in (INCOSE, 2011) rather than the logical sequence shown by (Scheithauer & Forsberg, 2013). Further, the Knowledge Growth Curve presented (Scheithauer, 2012) was found to be an effective way of communicating the difficulties experienced on this particular development project.

The comparison to best practices described in (Oppenheim et al., 2011) found a significant disconnect between how the project was conducted and what industry recognizes as being effective.

Product Development Improvements

This leads to the conclusion that more effective application of current Systems Engineering practice to the solar boat product development would have likely have improved project performance. Improvements to the development process follow the following themes:

- As stated in (Scheithauer) "iterations are the rule in systems engineering, not the exceptions." as such plans should explicitly include iterations (such as the prototype stage described in the improved Dynamic-V of Figure 9).

- An appreciation of the level of knowledge and skill of the team will enable an appropriate product development process to be selected. The Knowledge Growth Curve (Figure 7s) provided a simple way to visualize this for the project. Enabling future development to focus on how to increase initial skills and knowledge possessed by the team and increase skills during development.

- Decomposing the project by using the Basic-V defined by (Scheithauer & Forsberg, 2013) and mapping to the Development-V and Assurance-V forces a development team to actively think what work products are needed for the development and how to ensure they are tested sufficiently that the resulting developed system has adequate performance.

- Project progress was tracked by deviation from a fixed development plan which was written lacking considerable knowledge of the system at the start of the project. More meaningful metrics should be deployed to track project progress and make meaningful project plan updates over the course of the project.

- Modeling was used for some systems designs. Greater use of this across all systems will likely avoid rework cycles as the systems is integrated and found to have lacked capability.

- Between the individual sub teams, communications were poor. Greater use of defined work products and active discussion weekly surrounding these will likely improve this.

Future Research

As a result the following future research directions are proposed to better increase the adoption of Systems Engineering best practices including:

- The V-Model views presented in this paper were made retrospectively, how well they work during active development must be investigated, including software implementation.

- Modeling is required early in the development program to drive design decisions. Increasing the speed at which models can be created and utilized by the development of modelling libraries and the integration with existing engineering toolsets will further this.

- Measurement on software projects has been shown to improve predictability and development success (Jones, 2008). Application of metrics on real Systems Engineering projects is worthy of investigation.

- How to most effectively and efficiently teach System Engineering best practices to appropriate stakeholders of product development process such that real product development activities can be improved.

ACKNOWLEDGEMENTS

This research could not have been completed without the product development work undertaken as the solar boat project by students, professors and technical staff at the University of Tokyo and KTH Royal Institute of Technology.

University of Tokyo professors & technical stuff:
Katsuyuki Suzuki, Kenji Tanaka, Hideaki Murayama, Yoshihiro Tsuchiya, Makoto Kanai and Masaru Miyanaga

KTH professors:
Stefan Hallström, Jakob Kuttenkeuler and Anders Rosén

University of Tokyo students:
Daichi Ogawa, Daishi Sgawa, Tetsuo Tamaoki, Joshua Sutherland, Haruya Kamiyama, Ryutaro Matsumoto, Jialin Han, Yohei Kanno, Hiroaki Takoi, Yuya Kimura, Mengshi Zhu and Daiki Sakamoto

KTH students:
Urban Andern, Marten Kreiss Amberg, Paul Waldenstrom, Harsha Cheekamurthy, Linus Olin, Alexander Meldrum, Patrik Idstam, Durlabh Pande and Kwaku Gavor

Further information about the project is available at the following websites:
http://solarboat.sys.t.u-tokyo.ac.jp/
http://www.kthnavaldesign.se/kth-solar-2014

REFERENCES

INCOSE. (2004, June 14). *INCOSE - What is System Engineering?*. Retrieved December 7, 2014, from http://www.incose.org/practice/whatissystemseng.aspx

INCOSE. (2011). *Systems Engineering Handbook: A guide for system life cycle processes and activities* (v3.2.2 ed.). International Council of Systems Engineering.

Jones, C. (2008). *Applied Software Measurement: Global Analysis of Productivity and Quality* (3 edition.). New York: McGraw-Hill Osborne Media.

NASA. (2007). *NASA Systems Engineering Handbook*. NASA.

Oppenheim, B. W., Murman, E. M., & Secor, D. A. (2011). Lean Enablers for Systems Engineering. *Systems Engineering*, *14*(1), 29–55. doi:10.1002/sys.20161

Scheithauer, D. (2012). Managing Concurrency in Systems Engineering. *INCOSE International Symposium*, *22*(1), 2016–2030. doi:10.1002/j.2334-5837.2012.tb01453.x

Scheithauer, D., & Forsberg, K. (2013). V-Model Views. *INCOSE International Symposium*, *23*(1), 502–516. doi:10.1002/j.2334-5837.2013.tb03035.x

APPENDIX A

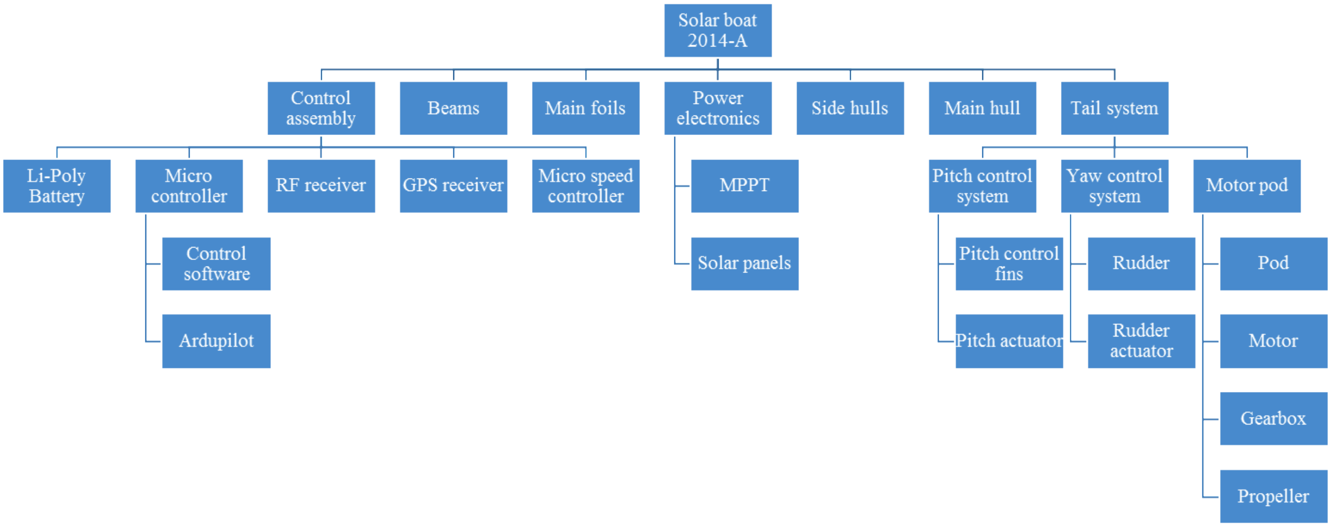

Figure A1: Solar boat "2014-A" Component structural decomposition hierarchy