

UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS

RAFAEL MOYA RODRIGUES PEREIRA

**Processos de Software para Implementações de Especificações  
Relacionadas ao Projeto Fórmula SAE/EESC/USP**

São Carlos  
2012



**RAFAEL MOYA RODRIGUES PEREIRA**

**PROCESSOS DE SOFTWARE PARA  
IMPLEMENTAÇÕES DE  
ESPECIFICAÇÕES RELACIONADAS  
AO PROJETO FÓRMULA  
SAE/EESC/USP**

Trabalho de Conclusão de Curso apresentado  
à Escola de Engenharia de São Carlos, da  
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em  
Eletrônica

ORIENTADOR: Professor Doutor Ivan Nunes da Silva

São Carlos  
2012

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,  
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS  
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

M436p

Moya Rodrigues Pereira, Rafael  
Processos de Software para Implementações de  
Especificações Relacionadas ao Projeto Fórmula  
SAE/EESC/USP / Rafael Moya Rodrigues Pereira;  
orientador Ivan Nunes da Silva. São Carlos, 2012.

Monografia (Graduação em Engenharia Elétrica com  
ênfase em Eletrônica) -- Escola de Engenharia de São  
Carlos da Universidade de São Paulo, 2012.

1. Engenharia de Software. 2. CAN. 3. Fórmula SAE.  
4. sensores. I. Título.

# FOLHA DE APROVAÇÃO

Nome: Rafael Moya Rodrigues Pereira

Título: "Processos de Software para Implementações de Especificações Relacionadas ao Projeto Fórmula SAE/EESC/USP"

*Trabalho de Conclusão de Curso defendido e aprovado  
em 28/11/2012,*

*com NOTA DEZ (10,0), pela Comissão Julgadora:*

**Prof. Associado Ivan Nunes da Silva (Orientador)**  
SEL/EESC/USP

**Prof. Dr. Danilo Hernane Spatti**  
SEL/EESC/USP

**M.Sc. Fernanda Maria da Cunha Santos**  
SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:  
Prof. Associado Homero Schiabel



Dedico este trabalho e a graduação para os meus pais, familiares, colegas de curso e aos amigos da equipe Fórmula SAE que participaram e me ajudaram a passar por esta etapa da minha vida.



## **Agradecimentos**

Aos meus pais, exemplos de fé, dedicação, honestidade e persistência, por toda a atenção, carinho e suporte que me dedicaram durante mais esta etapa de minha vida. A minha tia Encarnacion e a minha irmã sempre ao meu lado para apoiar, incentivar e motivar-me a continuar esta caminhada. A todos os familiares, que me deram força e motivação para sempre seguir em frente e sobrepor todas as barreiras que apareceram ao longo desses anos.

Aos colegas de equipe FSAE, em especial André Lui, Danilo Porto, José Ernesto por me motivarem e colobararem diretamente neste trabalho; Aos colegas de curso, em especial Michel Lacerda pelas suas orientações, Daniel Corral e Ciro Castro pelo incentivo as nossas participações em projetos extracurriculares, a Letícia Braga pela sua amizade e a Jussara Ribeiro que serviu de líder e exemplo por todos esses anos, aos companheiros de TCC Luciano Falqueto, Rafael Mattazio, e a Rafaela Peixoto pelas inúmeras trocas sobre formatação e redação de nossos trabalhos. Ao Professor Ivan pelo conhecimento, orientação e atenção fornecida durante a graduação e o trabalho.

E a todos que contribuíram de alguma forma para o sucesso deste trabalho e para a realização de um sonho.

Muito Obrigado!



## Resumo

Este trabalho faz parte de um projeto maior de desenvolvimento de um protótipo de competição universitária, FSAE, da equipe Fórmula SAE/EESC/USP da Escola de Engenharia de São Carlos, que consiste no desenvolvimento de softwares para controle, aquisição e transmissão de dados dos sensores do protótipo. Os sensores fornecem um sinal elétrico entre 0 e 5 V. Tais sinais serão então tratados pelo conjunto software/hardware desenvolvido pela equipe, em que neste caso, nós iremos estudar o projeto destes softwares, denominados POPA, PROA e Painel, conforme a posição do hardware no protótipo. Além de receber os sinais dos sensores, têm-se também o software que envia sinais para o atuador pneumático e para a ECU (Unidade de controle eletrônico do motor). Todos os sinais estão disponibilizados em um barramento CAN, que tem um nó transmissor que faz comunicação via RF com um programa em Labview, o qual exibe em tempo real a situação do protótipo na pista.

**Palavras-Chave:** Engenharia de Software, CAN, Fórmula SAE, sensores



## Abstract

This work is part of a bigger development project of a students competition prototype, known as FSAE, belonging to Fórmula SAE EESC/USP team from Escola de Engenharia de São Carlos. The project is based on control, aquisition and data transmission hardware/software from the sensors alocated in many places of the car. The sensors sends an eletric signal between 0-5V. This signals will be treated by the software/hardware developed by the team. In this case, we will study the project of this softwares, called by us POPA, PROA and Painel, according to the hardware's position in the car. Besides receiving the signals from the sensors,these softwares control a pneumatic actuador and communicate with the prototype's ECM (Eletronics Control Motor). The information is transmitted through the CAN bus, that has one transmissor node responsible for the RF communication with another software, this one developed in Labview, that shows real time data of the prototype during the race.

**Palavras-Chave:** Software Engineering, CAN, SAE Formula, sensors



## **Lista de Figuras**

Figura 1 - Quadro para controle das atividades de um projeto.....	24
Figura 2 - Modelo Real de Processo de Software .....	30
Figura 3 - Modelo Cascata.....	33
Figura 4 - Competição FSAE Brasil 2011 .....	36
Figura 5 - Prótotipo EX .....	37
Figura 6 - Organograma Equipe Formula SAE.....	38
Figura 7 - Organograma Equipe Eletrônica.....	39
Figura 8 – Exemplo de tela a ser exibida pelo software Tratamento de Dados .....	61
Figura 9 – Disposição dos Módulos .....	62
Figura 10 – Cronograma.....	78
Figura 11 – Projeto de arquitetura .....	78
Figura 12 - Fluxo de leitura de um dado analógico.....	79
Figura 13 - Fluxo da comunicação da ECM.....	79
Figura 14 - Unidade de Controle do Motor (ECM).....	80
Figura 15 - Fluxo da leitura de um dado digital (Atuadores).....	80
Figura 16 – Borboletas e válvula solenoide para troca de marcha .....	81
Figura 17 - Fluxo de leitura de um dado digital (Sensores) .....	81
Figura 18 - Sensor de velocidade .....	82
Figura 19 - Fluxograma procedimentos programação.....	85



## **Lista de Tabelas**

Tabela 1 – Atributos de software.....	25
Tabela 2 – Modelos de processo.....	29
Tabela 3 – Diferenças Engenharia de Software x demais Engenharias .....	40



## **Lista de Siglas**

FSAE : Fórmula SAE

SAE: Sociedade dos Engenheiros da Mobilidade

EESC: Escola de Engenharia de São Carlos

USP: Universidade de São Paulo



# Sumário

Agradecimentos .....	ix
Resumo .....	xi
Abstract .....	xiii
Lista de Figuras .....	xv
Lista de Tabelas .....	xvii
Lista de Siglas .....	xix
Sumário .....	xxi
Capítulo 1 – Introdução .....	23
1.1    Apresentação .....	23
1.2    Objetivos .....	25
1.3    Organização da Monografia .....	26
Capítulo 2 – Revisão de Literatura .....	29
2.1    Atividades de processo Engenharia de Software .....	30
2.1.1    Requisitos ou Especificação de software .....	30
2.1.2    Projeto de software (Código) .....	32
2.1.3    Testes .....	33
2.1.4    Manutenção .....	33
2.2    Conceitos de Fórmula SAE .....	35
2.2.1    A Competição FSAE .....	35
2.2.2    O Projeto EX e a Equipe EESC USP .....	37
2.2.3    O Projeto Eletrônica e a Equipe Eletrônica EESC USP .....	39
Capítulo 3 – Modelos de Relatórios Técnicos .....	43
3.1    Especificação Sistema Eletrônica FSAE .....	43
3.1.1    Objetivo .....	43
3.1.2    Escopo do Fornecimento .....	43
3.1.3    Arquitetura do Sistema Eletrônica FSAE .....	43
3.2    Relatório Técnico Softwares .....	44

3.2.1	Objetivo .....	44
3.2.2	Escopo do Fornecimento .....	44
3.2.3	Requisitos ou Especificação de software.....	44
3.2.3.1	Estudo da Viabilidade .....	44
3.2.3.2	Análise Econômica .....	45
3.2.3.3	Análise de Cronograma .....	45
3.2.4	Projeto de Software (Código).....	45
3.2.4.1	Projeto de arquitetura.....	45
3.2.4.2	Projeto de componente.....	46
3.2.4.3	Projeto de estruturas de dados.....	46
3.2.4.4	Projeto de algoritmo .....	46
3.2.5	Testes .....	46
3.2.5.1	Teste de Componente.....	46
3.2.5.2	Teste de Sistema .....	47
3.2.5.3	Teste de Aceitação .....	47
3.2.6	Manutenção .....	47
3.2.6.1	Manutenção Corretiva.....	47
3.2.6.2	Manutenção Adaptativa .....	47
Capítulo 4 – Resultados e Conclusões.....		49
4.1	Conclusões Gerais .....	49
4.2	Trabalhos Futuros .....	49
Bibliografia.....		51
Anexo 1 – Especificação Sistema Eletrônica FSAE .....		53
Anexo 2 – Relatório Técnico Softwares FSAE .....		65

# Capítulo 1 – Introdução

## 1.1 Apresentação

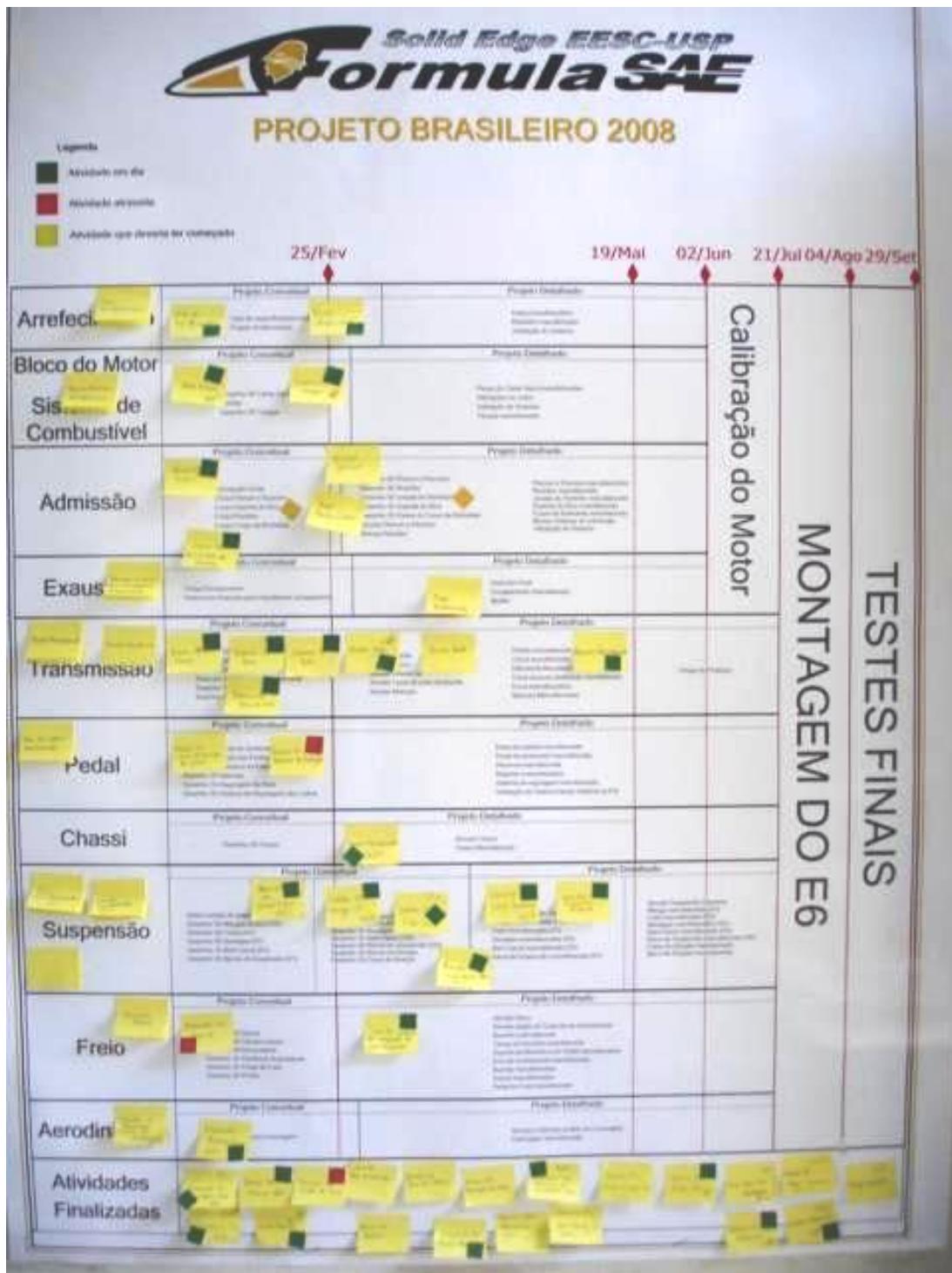
Nas últimas décadas o desenvolvimento tecnológico tem avançado rapidamente. Com a globalização, as necessidades de produtos específicos têm aumentado e com isso gerando a necessidade de adaptação dos projetos de engenharia. O desenvolvimento mecânico dos componentes percorreu um maior tempo; dessa forma, o desenvolvimento do software que irá controlar esse componente mecânico tem se tornado cada vez mais o diferencial de um projeto, sendo determinístico no funcionamento adequado do produto e, para tal desenvolvimento, o uso de métodos de projeto é de suma importância, consequentemente dando espaço a uma nova necessidade nos projetos, ou seja, controlar a qualidade, ditar métodos de testes, validação e etc, abrindo espaço para o surgimento e crescimento da Engenharia de Software, mas o que é engenharia de software?

Engenharia de Software é o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais “Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.” – Naur and Randell, [1].

Nos processos de desenvolvimento do protótipo de competição FSAE, consta nas regras determinadas pela SAE a possibilidade das equipes desenvolverem e investigar, além dos sistemas mecânicos, como freios, suspensão, chassi que influenciam o comportamento dinâmico do veículo, sistemas de *powertrain* e eletrônica, estes responsáveis por controle de todo fornecimento de potência e energia para movimentar o veículo, assim como sistemas de telemetria, aquisição de dados, controle e atuação responsáveis por fornecer aos outros integrantes (subsistemas) dados para validar e/ou desenvolver novas técnicas e produtos, visando o desenvolvimento dos estudantes das diversas engenharias aplicadas ao projeto.

O gerenciamento das atividades do projeto do protótipo é realizado num contexto geral, levando em conta alguns princípios de gestão de projetos, conhecidas técnicas de gerenciamento de projetos, contudo, ao lidar com o desenvolvimento de softwares para aquisição e tratamento de dados, controle de atuadores para troca de marcha, e sistema em tempo real de telemetria, devemos nos atentar à necessidade de tratamento diferenciado,

sabido que um software não pode ser manufaturado no sentido físico de entregar uma peça pronta para complementar um sistema de freios por exemplo. A Figura 1 apresenta um exemplo de quadro para controle das atividades de um projeto.



**Figura 1 - Quadro para controle das atividades de um projeto.**

## 1.2 Objetivos

Este trabalho propõe como objetivo principal aplicar conceitos de Engenharia de Software no projeto do software utilizado pela equipe Fórmula/SAE/EESC/USP para aquisição, tratamento e telemetria do veículo protótipo “EX” 2012.

De modo análogo ao projeto do sistema Estrutural, os projetos de software da Eletrônica devem seguir um método de produção e avaliação quanto aos seus atributos. Esses atributos são inerentes à função do software, e são comumente usados na avaliação de qualidade e refletem o comportamento perante (quanto) sua execução, estrutura e organização do código-fonte, bem como toda documentação referente ao projeto. O conjunto específico de atributos que você pode esperar de um sistema de software pode ser generalizado conforme Tabela 1 [2, 3].

**Tabela 1 – Atributos de software**

COMPONENTE	DESCRIÇÃO
Facilidade de manutenção	O software deve ser escrito de modo que possa evoluir para atender às necessidades de mudança dos clientes. É um atributo fundamental, pois a mudança de software é uma consequência inevitável de um ambiente de negócios em constante mutação.
Confiança	O nível de confiança do software tem uma série de características, incluindo confiabilidade, proteção e segurança. Um software confiável não deve causar danos físicos ou econômicos no caso de falha no sistema.
Eficiência	O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto, a eficiência inclui tempo de resposta, tempo de processamento, utilização de memória etc.
Usabilidade	O software deve ser usável, sem esforço excessivo, pelo tipo de usuário para o qual ele foi projetado. Isso significa que ele deve apresentar uma interface com o usuário e documentação adequadas.

A equipe EESC/USP/FSAE já possui tais sistemas e vem evoluindo o desenvolvimento e robustez dos softwares a cada novo protótipo; porém, com este trabalho

terá um padrão a ser seguido e posteriormente adequado com a evolução das necessidades, impactando em softwares mais confiáveis, de manutenção e evolução simplificada, e resultados melhores na competição.

Ciente que o time que trabalha no projeto de software da Equipe FSAE é formado por estudantes em formação com nível de maturidade incipiente em implementação de processos, e com o objetivo de implantar práticas de Engenharia de Software ao longo do projeto, a abordagem terá como foco definir cada atividade, gerar documentos técnicos de especificação de projeto e de documentação do projeto. [4]

Com este intuito, o trabalho reúne vários campos abordados durante o curso de graduação (circuitos eletrônicos, programação, instrumentação eletrônica) e também conhecimentos da Engenharia Mecânica em busca de atender os requisitos impostos para alcançar o objetivo do projeto do grupo de eletrônica a fim de prover informação aos diversos subsistemas da equipe em busca do melhor projeto de engenharia na competição FSAE Brasil.

### **1.3 Organização da Monografia**

A monografia se estrutura de acordo com a segmentação em capítulos proposta a seguir:

Capítulo 1: Introdução - Faz um apanhado geral do tema da monografia de modo a fornecer uma idéia abrangente do projeto.

Capítulo 2: Revisão de Literatura – Tem como objetivo apresentar o tema gerenciamento de projetos e engenharia de software, abordando alguns métodos e atividades usuais em projetos de software, assim como abordaremos brevemente sobre Fórmula SAE e suas atividades, contextualizando então este trabalho dentro do mesmo.

Capítulo 3: Modelos de Relatórios Técnicos – Este capítulo apresenta dois documentos básicos resultantes do estudo teórico de Engenharia de Software e de aplicações anteriores da equipe, assim como experiências adquiridas durante o estágio. Estes documentos serão autoexplicativos e tem-se por objetivo que sejam utilizados e aperfeiçoados com os projetos seguintes.

Capítulo 4: Resultados e Conclusões – Este capítulo apresenta as análises e os resultados obtidos com a implementação da metodologia. Aponta também possíveis melhorias a serem feitas futuramente e a conclusão do projeto.

Anexo 1: Documento de Especificação do Sistema Eletrônica FSAE

Anexo 2: Relatório Técnico (Softwares Proa, Popa e Painel)



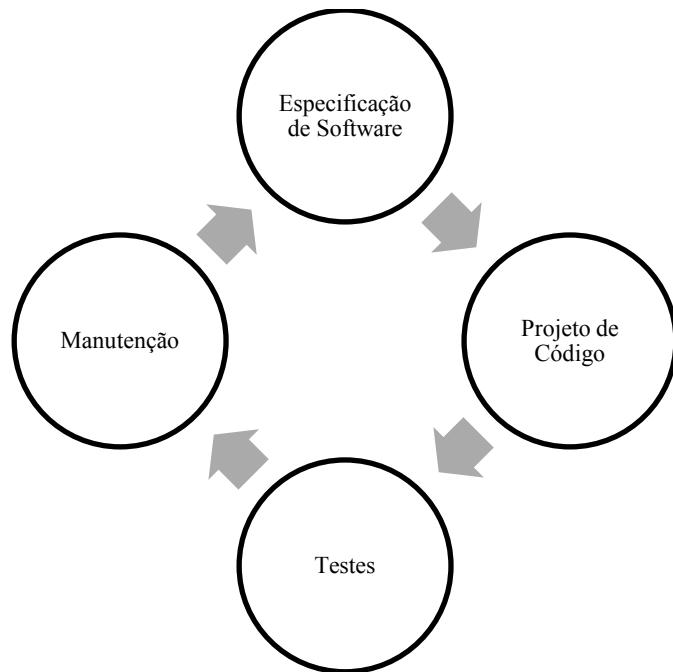
## Capítulo 2 – Revisão de Literatura

O desenvolvimento de software se orienta normalmente a partir de modelos, de representações abstratas e de processo de software. Na Tabela 2 apresentaremos alguns modelos genéricos de processo que podem ser ampliados e adaptados para criar processos mais específicos de engenharia de software [2].

**Tabela 2 – Modelos de processo**

MÉTODO	CARACTERÍSTICA
Modelo em cascata	Considera as atividades fundamentais do processo compreendendo especificação, desenvolvimento, validação e evolução, e as representa como fases de processos separadas, tais como especificação de requisitos, projeto de software, implementação, teste e assim por diante.
Desenvolvimento evolucionário	Esta abordagem intercala as atividades de especificação, desenvolvimento e validação. Um sistema inicial é desenvolvido rapidamente baseado em especificações abstratas. Este sistema é, então, refinado com as entradas do cliente para produzir um sistema que satisfaça as necessidades do cliente.
Engenharia de software baseada em componentes	Esta abordagem baseia-se na existência de um número significativo de componentes reutilizáveis. O processo de desenvolvimento do sistema enfoca a integração desses componentes, em vez de desenvolvê-los a partir do zero.

Em suma, na prática, são utilizados conceitos dos três modelos citados, como podemos notar, pelo modelo real de processos (Figura 2) durante o desenvolvimento dos softwares, partes podem ser reusadas, partes podem ter especificação simples e detalhada no início do projeto, outras ocorrerão em virtude de atualização das especificações ou alterações de hardware, sendo que no projeto final de software tenha sido empregado o uso em conjunto dos modelos, portanto a abordagem deste trabalho passa a ter foco nos processos em particular [5].



**Figura 2 - Modelo Real de Processo de Software**

## 2.1 Atividades de processo Engenharia de Software

Os processos de software divergem quanto à organização das quatro atividades básicas do processo – especificação, desenvolvimento, validação e evolução. No modelo em cascata, as atividades são organizadas sequencialmente, enquanto, no desenvolvimento evolucionário, elas são intercaladas. Como essas atividades realizadas dependem do tipo de software, pessoas e estruturas organizacionais envolvidas. Não existe uma forma certa ou errada de organizar essas atividades e essas especificamente seguem o mesmo formato, independente do modelo de processo [2].

### 2.1.1 Requisitos ou Especificação de software

A atividade de especificação de software pode ser considerada uma das etapas mais complexas devido à necessidade de compreender e definir os serviços necessários, restrições de operação e de desenvolvimento. A execução de um bom código e entrega de um software de qualidade depende do projeto inicial proveniente da especificação de software, sendo perceptível que erros na engenharia de requisitos conduzirão a falhas na implementação do sistema [5].

O resultado da especificação de software é um documento de requisitos, seguindo um processo com quatro fases principais para publicação final do documento para início do projeto de software por parte dos programadores e acertos financeiros por parte do departamento de venda/compra dos interessados, desta forma usualmente há dois níveis de detalhamento neste documento.

As quatro fases principais do processo de engenharia de requisitos:

- i. Estudo de viabilidade.
- ii. Elicitação e análise de requisitos.
- iii. Especificação de requisitos
- iv. Validação de requisitos.

Rotineiramente, a análise de requisitos ocorre desde o início do projeto até próximo da entrega do sistema ao cliente. Assim, as quatro fases ocorrem concomitantemente de forma a esclarecer os limites do projeto, pois com o andamento do projeto novas solicitações e novos obstáculos podem surgir gerando necessidade de adaptação, seja ela técnica, comercial ou de outra natureza. [2].

Padrão mais amplamente conhecido é o IEEE/ANSI 830-1998 (IEEE,1998). O padrão IEEE sugere a seguinte estrutura para os documentos de requisitos:

1. Introdução
  - 1.1 Propósito do documento de requisitos
  - 1.2 Escopo do produto
  - 1.3 Definições, acrônimos e abreviaturas
  - 1.4 Referências
  - 1.5 Visão geral do restante do documento
2. Descrição geral
  - 2.1 Perspectiva do produto
  - 2.2 Funções do produto
  - 2.3 Características dos usuários
  - 2.4 Restrições gerais
  - 2.5 Suposições e dependências
3. Requisitos específicos
4. Apêndices

## 5. Índice.

### **2.1.2 Projeto de software (Código)**

A atividade de desenvolvimento de software consiste em converter uma especificação de sistema em um sistema executável. Esta atividade envolve a definição da arquitetura, dos componentes, das interfaces do sistema, algoritmo entre outras específicas a cada projeto, geralmente ocorre nesta etapa um refinamento da especificação inicial do sistema.

Um projeto de software deve conter a descrição da estrutura de software a ser implementada, dos dados que são partes do sistema, das interfaces entre os componentes do sistema e, dos algoritmos usados. Todo processo do projeto é decomposto permitindo que os erros e as omissões em estágios anteriores sejam descobertos.

De acordo com o guia [6] esta atividade é de grande importância, pois é durante suas tarefas que podem ser detectados problemas de funcionamento e de especificação do software que podem comprometer seu uso e conclusão do mesmo. Caso estes problemas não fossem detectados nesta etapa, teríamos a conclusão do projeto do sistema comprometida e, em caso de possíveis ajustes, suas correções se tornariam bastante custosa e grande parte do trabalho de desenvolvimento seria perdido. Além disso, o projeto do software procura prover uma visão mais geral do sistema, permitindo-se calcular possíveis impactos das decisões tomadas durante o projeto e de cada decisão tomada no decorrer do projeto.

A saída de cada tarefa de projeto é uma especificação para o próximo estágio. Conforme o processo das tarefas é realizado, essas especificações tornam-se mais detalhadas, convergindo para especificações precisas dos algoritmos e estruturas de dados a serem implementados.

As tarefas resumidas desta atividade são as seguintes:

- i. Projeto de arquitetura.
- ii. Projeto de componente
- iii. Projeto de estruturas de dados
- iv. Projeto de algoritmo

### 2.1.3 Testes

As atividades de testes constituem os processos de software. Em alguns casos como no modelo cascata da Figura 3, é a última etapa antes da entrega ao cliente e, portanto é durante esta atividade que qualquer falha que tenha passado despercebida deve ser detectada, garantido que o sistema a ser entregue represente o que está definido na especificação de software [2, 5, 7]

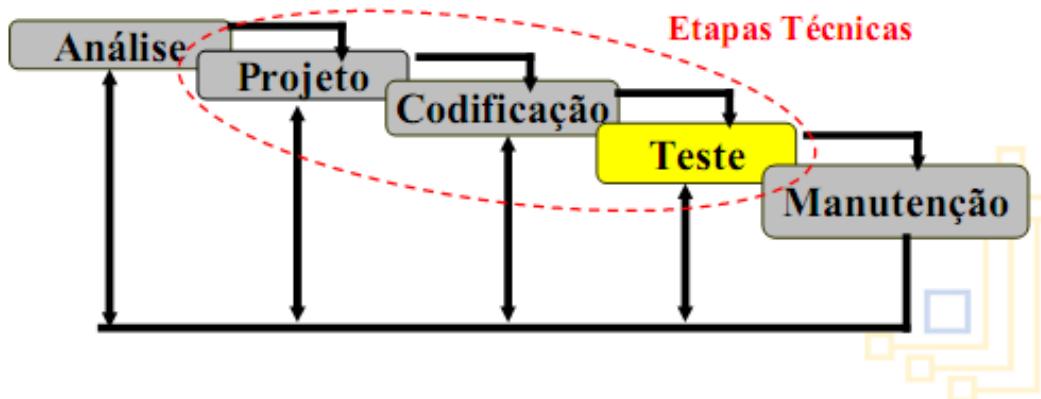


Figura 3 - Modelo Cascata

A fase de verificação e validação tem caráter revisor. E, desta forma, demanda em muitas ocasiões recursos em proporções maiores que a soma das outras etapas juntas, como de apresentar que o sistema está em conformidade com sua especificação para atender as exigências do cliente.

Os testes de software são distribuídos resumidamente em três etapas:

1. Teste de componente (ou unidade).
2. Teste de sistema.
3. Teste de aceitação.

Ressalta-se que as diversas literaturas existentes dizem que os testes não garantem que o software está correto, indicando apenas que a realização dos testes de modo rotineiro e planejado assegura o atendimento de requisitos mínimos de qualidade.

### 2.1.4 Manutenção

A atividade de manutenção é iniciada assim que o sistema de software é entregue para operação. No projeto corrente, esse momento ocorre quando o protótipo é lançado e uma primeira versão de todos os subsistemas contempla seu lançamento.

Diferentemente da manutenção de hardware, em que existe a fadiga por uso, desgaste do componente, falha ou defeito na fabricação, que pode ser feita com a troca ou reparo da peça que apresenta mau funcionamento, a manutenção de software, para correção de falhas/defeitos uma vez que em termos de software são sinônimos, uma vez que afetam a qualidade do produto, pode ocorrer mudanças simples para corrigir erros de codificação, mudanças mais extensas para corrigir erros de projetos ou melhorias significativas para corrigir erros de especificação ou para acomodar novos requisitos, conforme classificação abaixo.

Existem três tipos diferentes de manutenção de software, ou sejam:

1. *Manutenção para reparo de defeitos de software.(Corretiva).*
2. *Manutenção para adaptar o software a um ambiente operacional diferente (adaptativa).*
3. *Manutenção para adicionar funcionalidade ao sistema ou modifica-las (evolutiva).*

Os estudos realizados por [8] e abordados em [2] e [9] ressaltam, juntamente com a questão da manutenção evolutiva a questão da evolução do software, sendo esta decisão entre iniciar um novo projeto ou fazer a modificação para atender um novo requisito um processo não trivial, onde os custos envolvidos, a confiabilidade do software após manutenção e sua capacidade para mudanças futuras devem ser levadas em consideração.

Voltando estas discussões ao projeto da equipe constata-se claramente, como poderemos notar nos documentos anexos, que questões financeiras não devem influenciar o projeto negativamente, fazendo com que a decisão caminhe para o lado de confiabilidade e capacidade para mudanças futuras que nos leva a iniciar um novo projeto a cada novo protótipo da equipe, fazendo porém reuso de hardware, códigos e experiências anteriores devido à similaridade evolutiva entre os projetos.

## 2.2 Conceitos de Fórmula SAE

### 2.2.1 A Competição FSAE

Fórmula SAE é uma competição estudantil organizada pela SAE (Sociedade dos Engenheiros da Mobilidade). A primeira competição aconteceu em 1979, quando Mark Marshek, então na Universidade de Houston (Texas), entrou em contato com o Departamento de Relação Educacional da SAE para discutir a criação de uma variação da competição Mini Baja, dando início à nova categoria com o nome Mini Indy.

Tendo em vista o potencial do evento, Mike Best, Robert Edwards e John Telkamp, estudantes da Universidade do Texas em Austin, entraram em contato com Dr. Ron Matthews com uma ideia – que tal outro Mini-Indy, mas com algumas mudanças? Criaram novas regras, porém com maior flexibilidade à execução do projeto. O desejo era que essa nova competição levasse os carros próximo ao nível de engenharia, já que a competição de Baja era ótima para projetos estruturais, mas muitos estudantes também queriam a flexibilidade de trabalhar com os motores. As novas regras modificaram as restrições mínimas destes: seriam permitidos quaisquer motores 4 cilindros, desde que com potência limitada por uma restrição de 25.4mm na entrada de ar.

Com apoio total dos estudantes, Dr. Ron Matthews contatou o Departamento de Relação Educacional da SAE e colocou a nova ideia em ação. Para diferenciar este novo evento do Mini-Indy, era preciso um novo nome. Assim surge o Formula SAE, nome que reflete a natureza de corrida em pista do evento e o conteúdo avançado de engenharia.

O conceito por trás do Formula SAE é tal que uma empresa fictícia está contratando uma equipe de projeto para desenvolver um pequeno carro de corrida tipo Formula. O protótipo do carro de corrida será avaliado pelo seu potencial de produção em série. Para fins de marketing do projeto, o público alvo são os corredores de autocross amadores. Cada equipe deverá então projetar, construir e testar o protótipo seguindo uma serie de regras, cujo propósito é garantir a realização do evento com segurança e forçar a proposta de soluções inteligentes.

Formula SAE promove profissionalmente e excelência em engenharia, assim como engloba todos os aspectos da indústria automotiva, incluindo pesquisa, projeto, manufatura, testes, desenvolvimento, marketing, gerenciamento e finanças. Formula SAE leva os

estudantes para fora das salas de aula e permite a aplicação das teorias dos livros em casos reais de trabalho.

Atualmente, a competição se expandiu e inclui um grande número de eventos agregados. Internationalmente, participando da série Formula SAE, usando as regras Formula SAE Rules Copyright, há competições nos Estados Unidos, em duas localidades, Lincoln e Michigan, sendo este o maior e mais longo evento. Outras competições Formula são as seguintes

Formula SAE Australia

Formula SAE Brasil

Formula SAE Italia

Formula Student Reino Unido

Formula Student Alemanha

Formula SAE Japão

As informações descritas acima foram retiradas do site da SAE International [10] e seu objetivo é deixar o leitor familiarizado e ciente do contexto em que o trabalho foi realizado. A Figura 4 mostra dois integrantes recebendo os troféus em competição realizada em 2011.



Figura 4 - Competição FSAE Brasil 2011

## 2.2.2 O Projeto EX e a Equipe EESC/USP

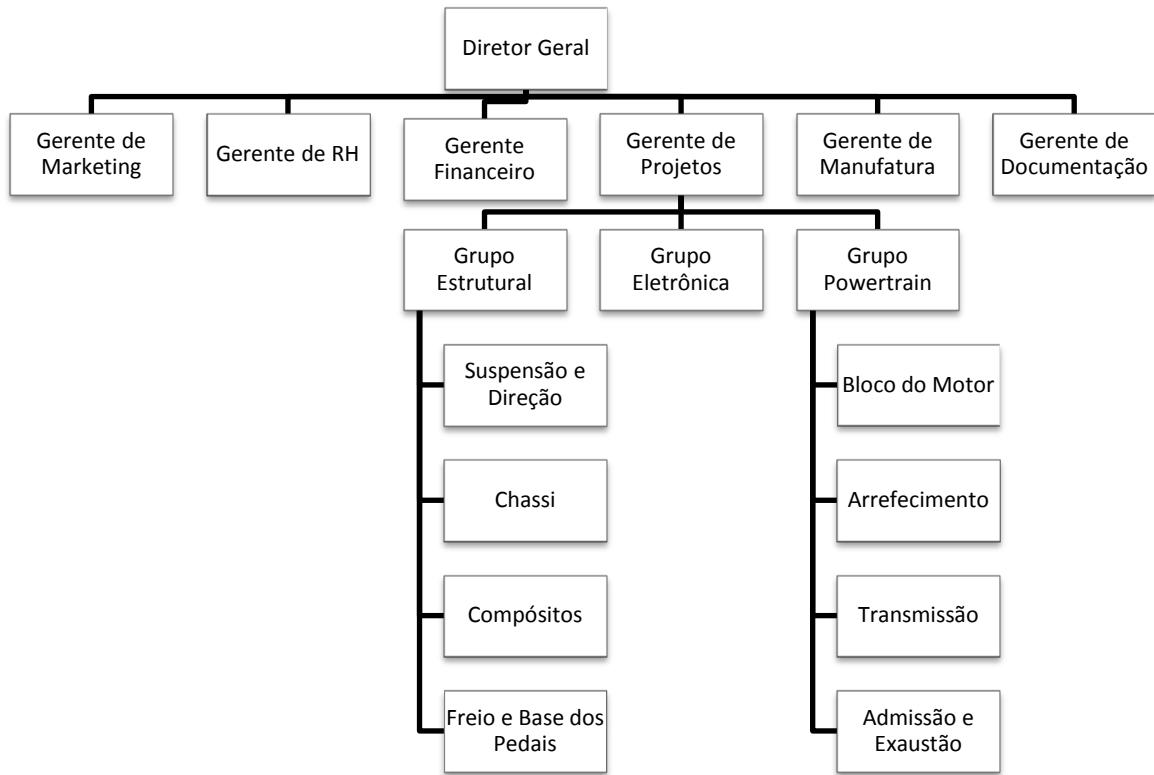
A Equipe EESC-USP de Formula SAE foi fundada em Junho de 2003 por alunos da Escola de Engenharia de São Carlos, alunos estes que em sua maioria eram ex-membros da Equipe EESC-USP de Mini-Baja. Neste ano a equipe está desenvolvendo seu 10º protótipo, denominado EX, para participar da 9ª Competição SAE Brasil – Petrobras de Formula SAE a ser realizada dos dias 30 de Novembro a 02 de Dezembro.



**Figura 5 - Protótipo EX**

A Figura 5 apresenta o protótipo durante testes realizados em Paulínia.

A equipe se divide em áreas técnica e administrativa, com subdivisão em sistemas e subsistemas, permitindo melhor divisão e controle das tarefas. Abaixo temos o organograma da equipe.



**Figura 6 - Organograma Equipe Formula SAE**

A gestão do projeto se inicia com a descrição das atividades com a construção de uma estrutura de decomposição de tarefas (EDT), posteriormente, a gestão elabora um cronograma para realização das tarefas, determinando prazos e datas limites para entrega dos produtos.

Durante a competição, cada equipe visa ter seu projeto aceito por um fabricante fictício. Os alunos devem trabalhar em equipe em todas as fases do projeto (desenvolvimento, construção, testes, promoção e operação), desenvolvendo um veículo que respeite as regras impostas pela organização. Cabe aos alunos viabilizar e gerenciar suporte financeiro para o projeto. Tudo deve ser feito sempre respeitando as prioridades acadêmicas.

### 2.2.3 O Projeto Eletrônica e a Equipe Eletrônica EESC/USP

Dentro do organograma exibido na Figura 6 nota-se o grupo da Eletrônica, responsável pelas tarefas de controle do motor, especificação e construção do chicote elétrico, desenvolvimento do software, controle para troca de marchas eletro-pneumatico. A equipe está subdividida conforme organograma abaixo (Figura 7).

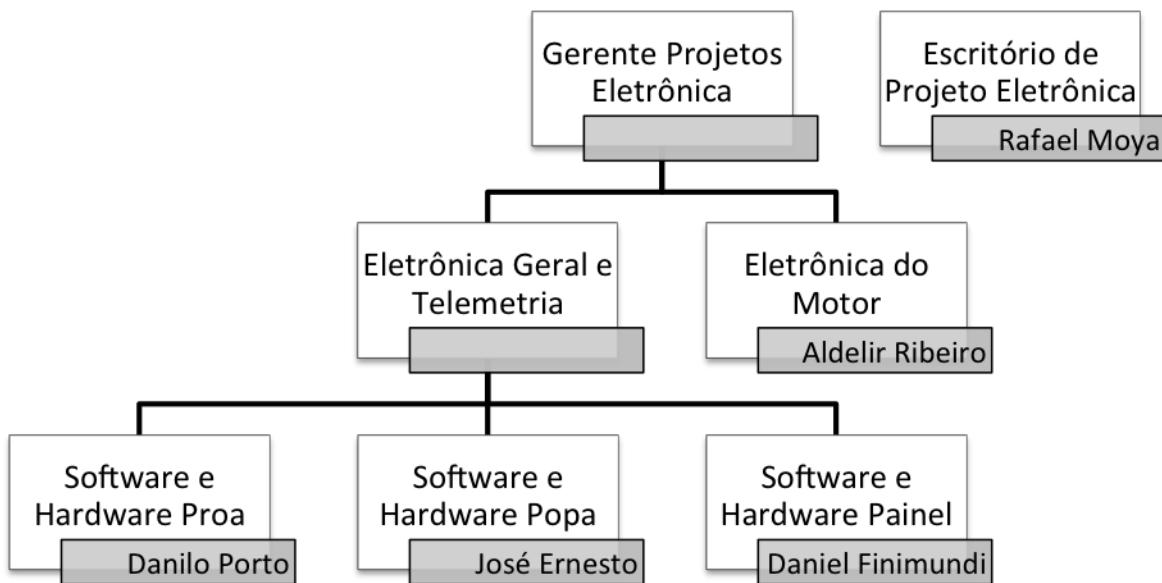


Figura 7 - Organograma Equipe Eletrônica

A gestão dos projetos deste subgrupo está alocada e segue o mesmo conceito e regras adotadas pela equipe como um todo. De certa forma, essa gestão tem se revelado suficientemente eficiente, visto os resultados alcançados pelo subgrupo nas competições, tendo ganhado por duas vezes o prêmio de inovação tecnológica. Pelo uso de telemetria em tempo real. E, posteriormente, com o uso da comunicação entre módulo seguindo protocolo CAN para elaboração do sistema de aquisição de dados e telemetria.

Ao analisarmos todas as atribuições referentes a este subgrupo, notamos as diferentes necessidades de acompanhamento, pois a gestão na engenharia de software é diferente da gestão das outras engenharias. Tais diferenças tornam esse gerenciamento um pouco mais complicado como podemos notar na

Tabela 3.

**Tabela 3 – Diferenças Engenharia de Software x demais Engenharias**

<b>CARACTERÍSTICA</b>	<b>DESCRIÇÃO</b>
Intangibilidade do produto	O gerente de um projeto de construção de um navio ou de um projeto de engenharia civil pode ver o produto que está sendo construído. Se o cronograma atrasa, o efeito sobre o produto é visível – partes da estrutura estarão, obviamente, não concluídas. O software é intangível. Não pode ser visto ou tocado. Os gerentes de projetos de software não podem ver o progresso. Eles contam com outras pessoas para produzir a documentação necessária para examinar o progresso.
Falta de processos padrão	Em áreas da engenharia com um longo histórico, o processo é experimentado e testado. O processo de engenharia de alguns tipos de sistema, como pontes e prédios, é bem compreendido. No entanto, os processos de software variam drasticamente de uma organização para outra. Embora nossa compreensão desses processos tenha se desenvolvido de modo significativo nos últimos anos, ainda não podemos prever, confiavelmente, quando determinado processo de software provavelmente apresentará problemas de desenvolvimento. Isso é especialmente verdadeiro quando o projeto de software é parte de um projeto mais amplo de engenharia de sistemas.
Projetos únicos	Projetos de software de grande porte são geralmente diferentes, de alguma forma, de projetos anteriores. Portanto, mesmo os gerentes que têm uma grande experiência podem considerar difícil prever problemas. Além disso, as rápidas mudanças de tecnologia em computadores e comunicações podem tornar a experiência do gerente obsoleta. As lições aprendidas em projetos anteriores podem não ser aproveitadas em novos projetos.

Estas diferenças justificam os atrasos, estouros de orçamento e a entrega do produto final fora do prazo em diversos projetos de software. Os projetos inovadores de engenharia, frequentemente, apresentam problemas com cronograma. Contando o caráter inovador presente nos sistemas de software e todas as dificuldades envolvidas, é de se surpreender que os projetos anteriores desenvolvidos pela equipe e que muitos projetos de software sejam entregues no prazo e dentro do orçamento! [2]



## **Capítulo 3 – Modelos de Relatórios Técnicos**

Esse capítulo descreve os modelos para documentação técnica do desenvolvimento dos softwares feitos, sendo descrito a necessidade e o porquê de cada item, sendo levado em conta o aprendizado teórico e padrões sugeridos ao longo do capítulo 2. Ao iniciar uma nova gestão e um novo projeto para a próxima competição, a equipe deve simular o pedido de um cliente a um fornecedor de software, especificando todos os dados, formas e necessidades referentes à eletrônica, neste focamos a parte de aquisição de dados para validação e desenvolvimento das partes mecânicas do protótipo.

Assim, definimos o primeiro documento a ser elaborado pela equipe, “Especificação Sistema Eletrônico FSAE” a ser realizado durante a etapa de especificação do sistema de eletrônica da equipe, listando os sensores, grandezas físicas que devem ser obtidas, formato de exibição na tela de monitoramento entre outras necessidades iniciais do projeto, e, um segundo documento a ser elaborado ao término do projeto denominado “Relatório Técnico Softwares” em que será relatada as etapas de especificação, projeto, testes e manutenção.

O resultado da aplicação destes modelos pode ser vistos nos anexos 1 e 2.

### **3.1 Especificação Sistema Eletrônica FSAE**

#### **3.1.1 Objetivo**

Descrever sucintamente a finalidade do documento, isto é, dizer a finalidade, metas, comportamento e desempenho desejado do produto do trabalho.

#### **3.1.2 Escopo do Fornecimento**

Descrever o quê deverá ser fornecido e o quê não será fornecido dentro desta especificação. Deve ser clara e abranger todos os itens de maneira mais específica possível, pois quando surgirem dúvidas no futuro, a gerência deste escopo permitirá garantir que o projeto realizou todo e somente o trabalho necessário para atingir seu objetivo.

#### **3.1.3 Arquitetura do Sistema Eletrônico FSAE**

Descrever uma proposta de arquitetura, em termos de módulos (hardware), que atenda as necessidades descritas na seção objetivo, para possibilitar o desenvolvimento dos softwares necessários.

## **3.2 Relatório Técnico Softwares**

Este documento deve ser finalizado e entregue à gerência de projeto da equipe assim que o software for aprovado. Tal documento poderá ser usado nas provas de Design Judge e Apresentação, com objetivo de auxiliar e agregar ao trabalho já realizado pela equipe.

### **3.2.1 Objetivo**

Descrever sucintamente a finalidade do documento, isto é, dizer a finalidade, metas, comportamento e desempenho desejado do produto do trabalho.

### **3.2.2 Escopo do Fornecimento**

Descrever o quê está sendo fornecido e o quê não foi fornecido dentro deste projeto de software conforme as especificações de software.

### **3.2.3 Requisitos ou Especificação de software**

#### **3.2.3.1 Estudo da Viabilidade**

Como em qualquer novo processo, engenharia de requisitos também deve começar com o estudo de viabilidade, cujo documento para estudo deve ser uma especificação inicial como definida na Seção 3.1. O resultado deste estudo definirá qual a melhor forma e se a empresa deve prosseguir com a negociação. Esta atividade não se aplica diretamente às atribuições da equipe FSAE, porém é interessante o estudo devido a aplicação parcial da teoria, onde pode ser aprovado o prosseguimento parcial da especificação inicial. Normalmente, deve-se tentar concluir um estudo de viabilidade em duas ou três semanas [2].

- a) Viabilidade Operacional

Nesta seção, devem ser detalhados os estudos de viabilidade operacional considerando as questões de recursos financeiros e humanos, de performance, de eficiência e nível de dificuldade de implementação.

b) **Viabilidade Técnica**

É uma avaliação da praticidade da solução técnica específica e a disponibilidade dos recursos técnicos e dos especialistas.

### **3.2.3.2 Análise Econômica**

a) **Custo de Aquisição**

Descrever o custo de aquisição leva em consideração a aquisição de um sistema em pleno funcionamento que atenda diretamente os requisitos e necessidades levantadas nas etapas anteriores, caso em que a equipe adquire produtos no mercado.

b) **Custos para desenvolvimento**

Descrever o custo de desenvolvimento deve levar em consideração a aquisição de todos os componentes necessários para desenvolvimento do um sistema completo e em pleno funcionamento que atenda diretamente os requisitos e necessidades levantadas nas etapas anteriores, caso em que a equipe desenvolve os produtos.

### **3.2.3.3 Análise de Cronograma**

Deve ser elaborado um cronograma com as atividades do processo de desenvolvimento do software contendo marcos de entregas, tarefas a serem realizadas, divisão de trabalho na equipe e acompanhamento das atividades.

## **3.2.4 Projeto de Software (Código)**

Descrever todo o projeto de software e suas etapas como definidas abaixo

### **3.2.4.1 Projeto de arquitetura**

Sistemas grandes são sempre decompostos em subsistemas que fornecem algum conjunto de serviços relacionados. O processo inicial de projeto consiste em identificar

esses subsistemas e estabelecer um framework para controle e comunicação dos subsistemas [2].

#### **3.2.4.2 Projeto de componente**

O projeto em nível de componente para software é equivalente a um conjunto de desenhos detalhados para cada cômodo de uma casa. O projeto em nível de componente de software descreve completamente os detalhes internos para todos os objetos de dados locais e detalhes algorítmicos para todo o processamento que ocorre em um componente e em uma interface que dá acesso a todas as operações do componente [5].

#### **3.2.4.3 Projeto de estruturas de dados**

O projeto de dados consiste na interpretação de objetos de dados em estruturas de dados para os componentes de software e, quando necessário, uma arquitetura de banco de dados para a aplicação, organizando o arquivamento destes dados e facilitando assim uma futura consulta e extração do mesmo [5].

#### **3.2.4.4 Projeto de algoritmo**

O projeto de algoritmo consiste em estudo e definição das lógicas de procedimentos, cálculos, taxas de amostragem, sequenciamento, determinação da linguagem e software para escrita do código, sua respectiva organização e documentação.

### **3.2.5 Testes**

Descrever os métodos e resultados obtidos durante os testes, assim como problemas e soluções encontradas.

#### **3.2.5.1 Teste de Componente**

Descrever os testes de componente individuais do sistema, expondo os defeitos encontrados. Na maioria dos sistemas, os desenvolvedores são responsáveis pelo teste de componente.

### **3.2.5.2      Teste de Sistema**

Descrever os testes de sistema realizados, teste esse que envolve a integração de dois ou mais sistemas que implementam funções características e, depois, o teste desse sistema todo integrado, onde erros são encontrados e soluções apontadas. A descrição destes testes possibilitará maior facilidade para manutenção e entendimento do código [5].

### **3.2.5.3      Teste de Aceitação**

Descrever o teste de aceitação e o real funcionamento para aprovação do sistema, descrever os pontos que devem ser corrigidos a partir da etapa de manutenção.

## **3.2.6      Manutenção**

Descrever como as atividades de manutenção podem vir a ocorrer, quais as possíveis causas e soluções para garantir manutenibilidade. As manutenções podem incluir correções, melhorias ou adaptações do software causada por mudanças no ambiente e nos seus requisitos ou especificações funcionais. (ISO/IEC 9126, 2003)

O conceito de manutenção por vezes se assimila com evolução do software, podendo por vezes ser considerado um novo produto, pois não estava previsto no escopo. Dentro do projeto Formula SAE, consideraremos manutenção toda atividade realizada após a primeira entrega em funcionamento do projeto.

### **3.2.6.1      Manutenção Corretiva**

Descrever todas as manutenções corretivas. Visando correção de defeitos de software, realizadas até aceitação do software.

### **3.2.6.2      Manutenção Adaptativa**

Descrever todas as manutenções adaptativas realizadas, visando adaptar o software à mudança de hardware, até aceitação do software.



## **Capítulo 4 – Resultados e Conclusões**

### **4.1 Conclusões Gerais**

Neste trabalho foi realizado o estudo para aplicação de métodos e conceitos pertinentes à Engenharia de Software, com intuito de aplica-los nos processos desenvolvidos pela equipe de eletrônica Fórmula SAE/EESC/USP.

Também foi realizado um comparativo entre as atividades de gerenciamento relacionado a projeto de carro e de software, demonstrando a real necessidade e importância do estudo e aplicação de Engenharia de Software nos projetos atuais em que temos o desenvolvimento de algum software, seja ele parte ou produto único do projeto.

A aplicação dos estudos ocorre com a determinação dos modelos apresentados no capítulo 3, que deram origem aos anexos 1 e 2 apresentados. Tal capítulo apresenta de forma sucinta os principais itens a serem descritos e observados durante o projeto de software e sugere uma documentação apropriada visando prover informações para todos os futuros envolvidos no projeto Formula SAE, facilitando assim qualquer trabalho de manutenção e/ou novos desenvolvimentos.

Portanto, considerando o caráter acadêmico e aplicado (profissional), percebe-se claramente que os objetivos foram alcançados, uma vez que conceitos de engenharia de software foram aplicados durante o projeto, isto é, tarefas anteriormente executadas voluntariamente a partir de experiências individuais passaram a ter um escopo bem definido, que poderá e deverá evoluir conforme as necessidades de projeto.

### **4.2 Trabalhos Futuros**

O projeto abre possibilidades para inúmeras melhorias e revisões. Logo de princípio, os projetos devem se iniciar com algum conhecimento sobre a Engenharia de Software e suas diversas aplicações. Este projeto deu enfoque ao estudo inicial que ocorreu paralelamente ao projeto de software da equipe. Desta forma, não houve, por exemplo, um maior aproveitamento dos estudos realizados na parte de especificação de requisitos, parte primordial para um bom resultado final. Tal etapa foi realizada nos mesmos moldes dos projetos anteriores.

Um novo projeto com proposta similar ao deste poderia sugerir o uso dos modelos apresentados para atingir níveis de exigência dentro do padrão ISO9126 de qualidade de software, incrementando o novo conhecimento e experiência dos desenvolvedores de software do projeto;

Pode-se também implementar a aplicação visando controle de custos, através de apresentação de diversas propostas de arquiteturas a partir de uma especificação de requisitos mais ricos, sendo este um trabalho que envolvesse a participação de toda equipe responsável pelo projeto, fato este que não ocorreu neste estudo devido ao desconhecimento da possibilidade de apresentações e trabalhos de conclusão de curso em grupo.

A realização de projetos com esta finalidade colabora para o desenvolvimento acadêmico e profissional dos envolvidos, e contempla a formação do estudante com o entendimento da relação entre cada atividade até a entrega do produto final, além de prover à nação, profissionais com maior conhecimento sobre engenharia de software e ajudando a evolução da qualidade dos softwares desenvolvidos no Brasil.

## Bibliografia

- [1] P. Naur and B. Randell, (Eds.). Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO (1969).
- [2] SOMMERVILLE, Engenharia de software, 8<sup>a</sup> edição, São Paulo: Pearson Addison-Wesley, 2007.
- [3] ISO, “ISO/IEC 9126: Engenharia de software - Qualidade de produto,” 2003.
- [4] K. Weber, E. Araújo, A. R. Rocha, K. Oliveira, A. C. Rouiller, C. Gresse von Wangenheim, R. Araújo, C. Salviano, C. Filipak Machado, D. Scalet, O. Galarraga, M. Pecegueiro Amaral e D. Yoshida, “Melhoria de Processo do Software Brasileiro (MPS.BR) um Programa Mobilizador,” [Online]. Available: <http://www.softex.br/portal/softexweb/uploadDocuments/26.pdf>. [Acesso em 04 Outubro 2012].
- [5] R. S. Pressman, Engenharia de Software, São Paulo: MCGRAW-HILL, 2006.
- [6] SWEBOK, “THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC - IEEE,” GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE, 2004.
- [7] E. Y. Nakagawa, Uma Contribuição ao Projeto Arquitetural de Ambientes de Engenharia de Software, São Carlos, 2006.
- [8] M. M. PADUELLI, “MANUTENÇÃO DE SOFTWARE: PROBLEMAS

TÍPICOS E DIRETRIZES PARA UMA DISCIPLINA ESPECÍFICA,”  
São Carlos, 2007.

- [9] S. L. Pfleeger, Engenharia de Software: teoria e prática, São Paulo:  
Prentice Hahll, 2004.
- [10] “Formula SAE International,” SAE International, [Online]. Available:  
<http://students.sae.org/competitions/formulaseries/about.htm>. [Acesso em  
03 Outubro 2012].

## **Anexo 1 – Especificação Sistema Eletrônica FSAE**



UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS

EQUIPE FORMULA SAE EESC USP

**Especificação Sistema Eletrônica FSAE**

São Carlos  
2012



## **Sumário**

1 Objetivo .....	59
2 Escopo de fornecimento .....	61
3 Arquitetura do Sistema Eletrônica FSAE.....	63



## **Objetivo**

Estabelecer as diretrizes e os critérios para fornecimento da plataforma de software e aplicativos computacionais de aquisição de dados referentes ao sistema Eletrônica FSAE do projeto Formula SAE EESC USP, protótipo com objetivo de participar da competição universitária SAE Brasil a ser realizada em novembro de 2012 na cidade de Piracicaba-SP, com finalidade de medição e análise de desempenho e rendimento do protótipo.



## Escopo de fornecimento

- **Sistema Aquisição de Dados:** sistema de software em LabView que irá adquirir os dados provenientes dos módulos Proa e Popa e possibilitar monitorá-los, gravar em planilhas de dados e exibir em uma tela de computador de forma visual ao cliente, com valores em km/h, m/s<sup>2</sup>, entre outros a serem detalhados (Figura 8).



Figura 8 – Exemplo de tela a ser exibida pelo software Tratamento de Dados

- **Módulo Popa:** Sistema de software em linguagem C desenvolvido com microcontroladores PIC18F2680 que deverá aquisitar os dados de todos os sensores da parte traseira do carro, atuar para realização das trocas de marchas, assim como converter os dados aquisitados em grandezas físicas e enviá-los via CAN para o módulo Painel
- **Módulo Proa:** Sistema de software em linguagem C desenvolvido com microcontroladores PIC18F2680 que deverá aquisitar os dados de todos os sensores da

dianteira do protótipo, comunicar com a ECM obtendo os dados desejados, convertê-los em grandezas físicas e enviá-los via CAN para o módulo Painel

- **Módulo Painel:** Sistema de software em linguagem C desenvolvido com microcontroladores PIC18F2680 com finalidade de exibir as informações mais relevantes ao piloto e, principalmente, enviar os dados para o transceptor, possibilitando a captação dos dados no software “Sistema Aquisição de Dados”.
- **Módulo ECM:** Sistema de hardware e software utilizado para controle dos parâmetros necessários para calibração e funcionamento adequado do motor.

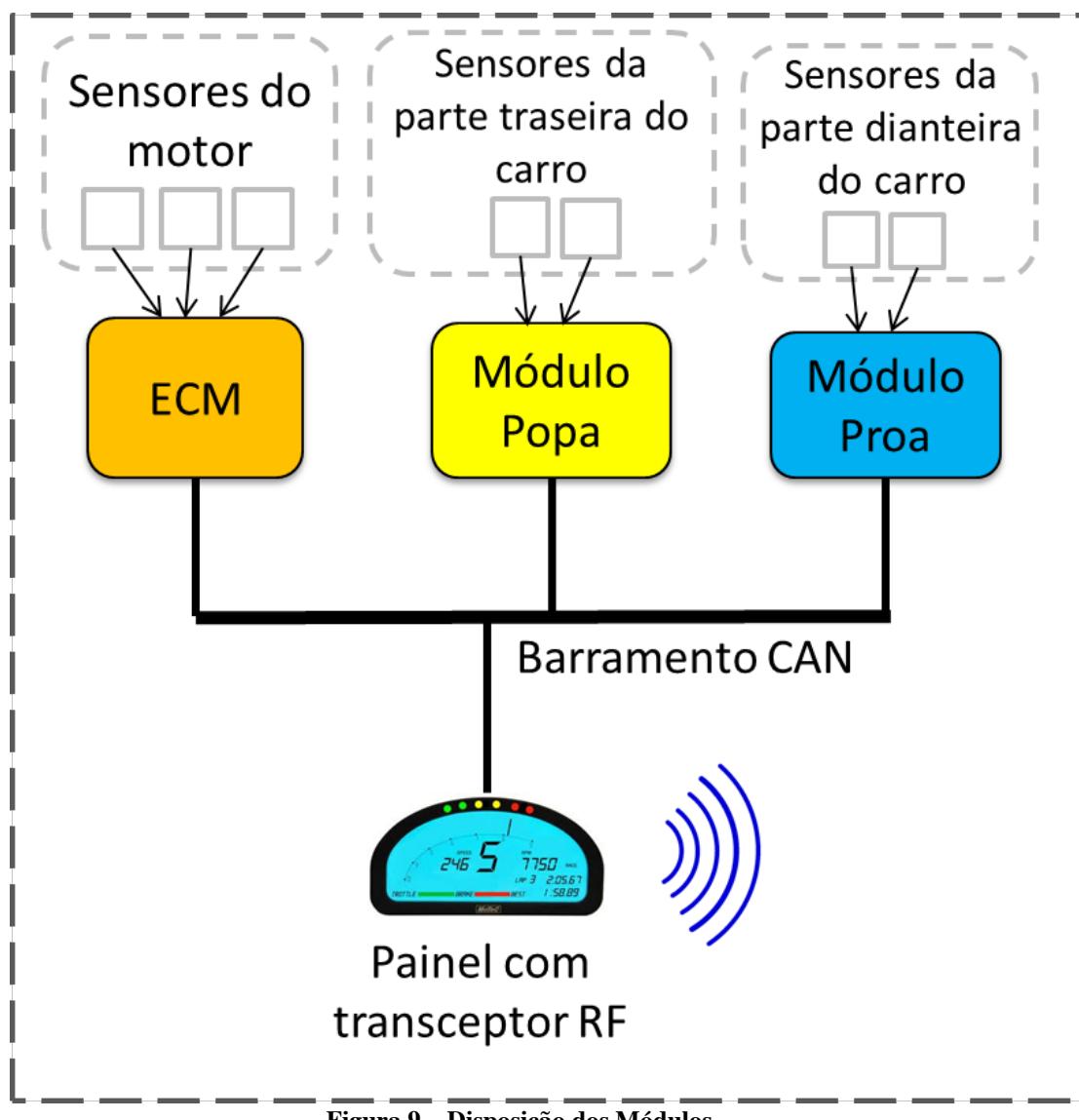


Figura 9 – Disposição dos Módulos

## Arquitetura do Sistema Eletrônica FSAE

A arquitetura do Sistema Eletrônica FSAE deverá ser composta pelos módulos popa, proa, painel e ECM (unidade de controle do motor), captando sinais dos sensores listados abaixo:

- 4 sensores para medir velocidade.
- 4 sensores de pressão da linha do freio
- Temperatura do ar
- Pressão do óleo do cárter
- Acelerômetro de 2 eixos
- RPM do motor
- Temperatura da água
- Tensão da bateria
- Relação ar/combustível na exaustão (Sensor lambda)
- Posição da borboleta (acelerador)
- Pressão do ar no coletor de admissão (MAP)
- Curso do pedal de freio e embreagem
- Curso da direção



## **Anexo 2 – Relatório Técnico Softwares FSAE**



UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS

EQUIPE FORMULA SAE EESC USP

**Relatório Técnico Software FSAE**

São Carlos  
2012



## **Lista de Figuras**

Figura 10 – Cronograma.....	78
Figura 11 – Projeto de arquitetura.....	78
Figura 12 - Fluxo de leitura de um dado analógico.....	79
Figura 13 - Fluxo da comunicação da ECM.....	79
Figura 14 - Unidade de Controle do Motor (ECM).....	80
Figura 15 - Fluxo da leitura de um dado digital (Atuadores).....	80
Figura 16 – Borboletas e válvula solenoide para troca de marcha .....	81
Figura 17 - Fluxo de leitura de um dado digital (Sensores) .....	81
Figura 18 - Sensor de velocidade .....	82
Figura 19 - Fluxograma procedimentos programação.....	85



## **Lista de Tabelas**

Tabela 1 - Mensagens .....	83
Tabela 2 - Normas, resoluções e intervalo dos principais dados.....	84

## **Lista de Siglas**

## Sumário

Lista de Figuras .....	69
Lista de Tabelas .....	71
Lista de Siglas .....	72
Sumário .....	73
Capítulo 1 – Relatório Técnico Softwares .....	75
1.1    Objetivo .....	75
1.2    Escopo do fornecimento .....	75
1.3    Requisitos ou Especificação de software .....	76
1.3.1    Estudo da Viabilidade .....	76
1.3.2    Análise Econômica .....	77
1.3.3    Análise de Cronograma .....	77
1.4    Projeto de Software (Código) .....	78
1.4.1    Projeto de arquitetura .....	78
1.4.2    Projeto de componente .....	79
1.4.3    Projeto de estruturas de dados .....	82
1.4.4    Projeto de algoritmo .....	83
1.5    Testes .....	85
1.5.1    Teste de Componente .....	85
1.5.2    Teste de Sistema .....	85
1.5.3    Teste de Aceitação .....	85
1.6    Manutenção .....	86
1.6.1    Manutenção Corretiva .....	86
1.6.2    Manutenção Adaptativa .....	86
Capítulo 2 – Resultados e Conclusões .....	87
2.1    Conclusões Gerais .....	87



# Capítulo 1 – Relatório Técnico Softwares

Apresentamos neste documento o desenvolvimento e execução do projeto dos softwares da equipe de eletrônica FSAE relatando as atividades realizadas ao longo do processo.

## 1.1 Objetivo

Este relatório tem como objetivo iniciar uma cultura com adoção de Engenharia de Software dentro da equipe de desenvolvedores e permitir evolução dos mesmos e dos softwares produzidos pela equipe, indiretamente colaborando com todo o restante da equipe em busca dos melhores projetos e resultados nas competições.

Neste documento será abordado o escopo do fornecimento, estudo de viabilidade, projeto de software, testes e manutenção de modo a descrever os procedimentos realizados pela equipe durante o projeto 2012.

## 1.2 Escopo do fornecimento

O projeto de desenvolvimento segue o descrito na “Especificação Sistema Eletrônico FSAE” e procura atender as necessidades para desenvolvimento e evolução dos outros subsistemas do protótipo. A seguir está descrito os módulos do fornecimento.

*Sistema Aquisição de Dados:* sistema de software em LabView que irá adquirir os dados provenientes dos módulos Proa e Popa e possibilitar monitorá-los, gravar em planilhas de dados e exibir em uma tela de computador de forma visual ao cliente, com valores em km/h, m/s<sup>2</sup>, entre outros a serem detalhados.

*Módulo Popa:* Sistema de software em linguagem C desenvolvido com microcontroladores PIC18F2680 que deverá aquisitar os dados de todos os sensores da parte traseira do carro, atuar para realização das trocas de marchas, assim como converter os dados aquisitados em grandezas físicas e enviá-los via CAN para o módulo Painel

*Módulo Proa:* Sistema de software em linguagem C desenvolvido com microcontroladores PIC18F2680 que deverá aquisitar os dados de todos os sensores da dianteira do protótipo, comunicar com a ECM obtendo os dados desejados, convertê-los em grandezas físicas e enviá-los via CAN para o módulo Painel

*Módulo Painel:* Sistema de software em linguagem C desenvolvido com microcontroladores PIC18F2680 com finalidade de exibir as informações mais relevantes ao piloto e, principalmente, enviar os dados para o transceptor, possibilitando a captação dos dados no software “Sistema Aquisição de Dados”.

## **1.3 Requisitos ou Especificação de software**

### **1.3.1 Estudo da Viabilidade**

#### **c) Viabilidade Operacional**

Disposição de apoiadores, novos e antigos, recursos financeiros, recursos humanos, dificuldades de implementação.

Do ponto de vista financeiro, o projeto se tornou viável devido ao apoio dos patrocinadores. Diversas empresas forneceram gratuitamente seus produtos para desenvolvimento do projeto.

Do ponto de vista de recursos, a equipe tem pessoal suficiente, e sempre buscando aprender para sanar as dificuldades e eventuais problemas, uma vez que toda ela é composta por estudantes. Deve ser considerado o lado acadêmico do projeto neste estudo de viabilidade.

Do ponto de vista das dificuldades de implementação, leva-se em conta novamente o caráter acadêmico da equipe e implementa-se o software atendendo todas as necessidades descritas na especificação.

Do ponto de vista de perfomance e eficiência, projeta-se os softwares modulares, para que o funcionamento deles seja independente.

#### **d) Viabilidade Técnica**

Todos os requisitos são atendidos em termos técnicos, pois é possível alcançar e prover todas as informações requisitados. Atenta-se também para buscar contato de patrocínio com os fabricantes dos sensores necessários para aquisição dos dados especificados. A disponibilidade de especialista não é considerada para definição deste estudo, uma vez que existe a possibilidade de contar com apoio de professores com qualificação para ajudar o desenvolvimento e aprendizado dos membros da equipe de eletrônica.

### **1.3.2 Análise Econômica**

No inicio do projeto a equipe de eletrônica lista juntamente dos grupos estrutural e powertrain os sensores necessários. Partindo desta lista é procurada as empresas fabricantes destes sensores em busca de parcerias e posteriormente a aquisição de sensores no mercado de acordo com as diretrizes do projeto FSAE.

#### **c) Custo de Aquisição**

Custo de aquisição leva em consideração a aquisição dos componentes necessários para montagem de um sistema completo para pleno funcionamento.

Os componentes comprados foram placas de cobre, conectores, componentes elétricos (resistores, capacitores, etc), Doados: microcontrolador, sensores, chicote elétrico.

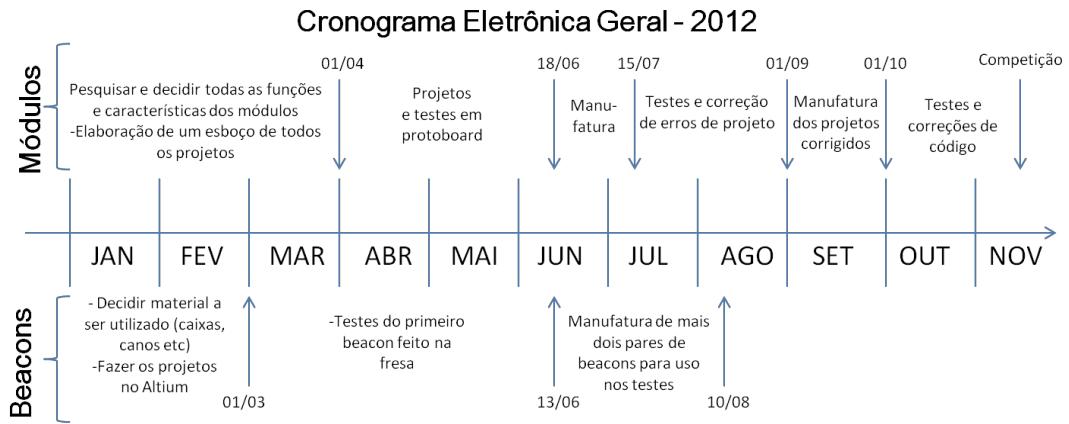
#### **d) Custos para desenvolvimento**

Custo de desenvolvimento leva em consideração a aquisição de todos os componentes necessários para desenvolvimento do um sistema.

Os componentes usados foram: softwares Altium, Mplab, Circuit CAM, computadores, fontes de tensão, osciloscópio, protoboards, equipamentos para soldagem.

### **1.3.3 Análise de Cronograma**

Elaboração de um cronograma com as atividades do processo de desenvolvimento do software contendo marcos de entregas, tarefas a serem realizadas, divisão de trabalho na equipe e acompanhamento das atividades, como segue na Figura 10.

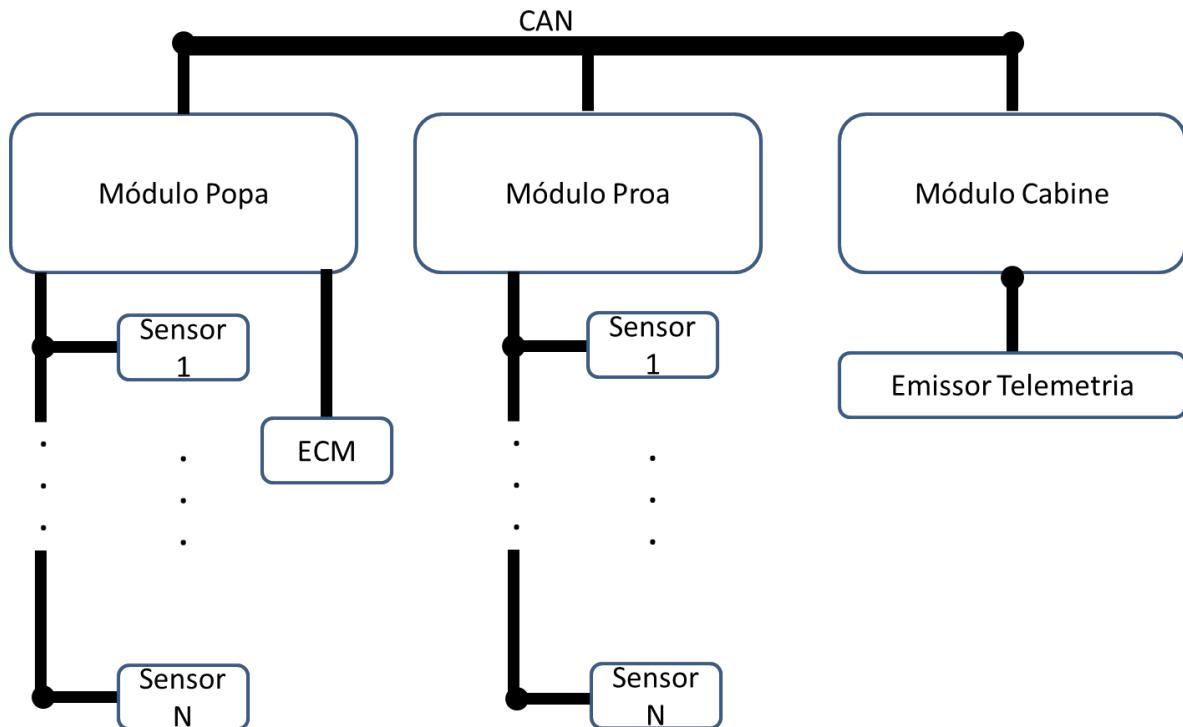


**Figura 10 – Cronograma**

## 1.4 Projeto de Software (Código)

### 1.4.1 Projeto de arquitetura

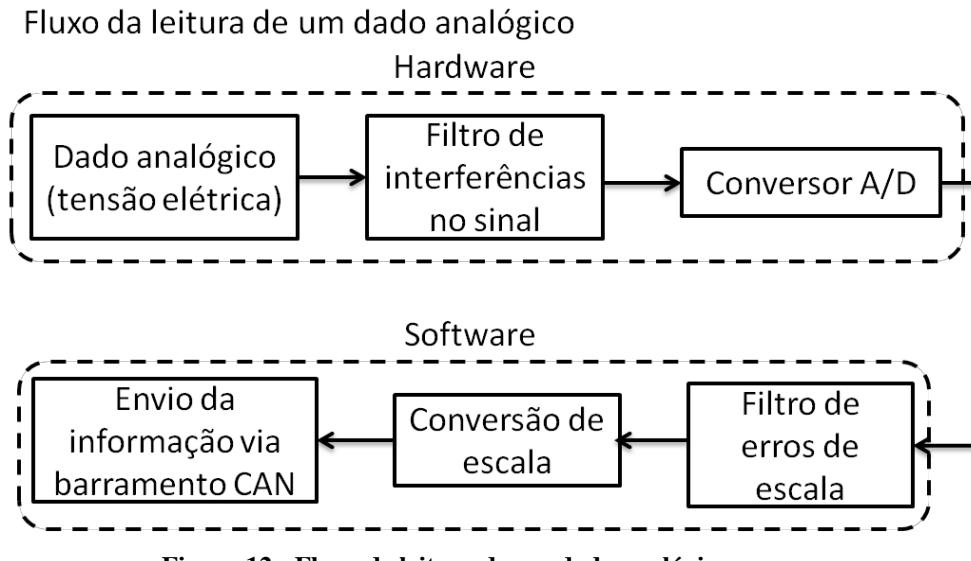
O projeto de arquitetura leva em conta a disposição física dos módulos no carro, e divisão das tarefas para facilitar e permitir um sistema modular. Desta forma segue-se a Figura 11 representativa da arquitetura projetada.



**Figura 11 – Projeto de arquitetura**

### 1.4.2 Projeto de componente

A estrutura do projeto de componente é descrita pelos fluxogramas a seguir, mostrando o fluxo dos dados desde que é captado no sensor até sua disponibilização na rede.



A Figura 12 mostra a leitura de um dado analógico, dentre eles temos pressão do freio, curso do pedal, curso da direção, curso da embreagem, pressão do óleo e do freio e acelerômetro, que seguem o mesmo modelo.

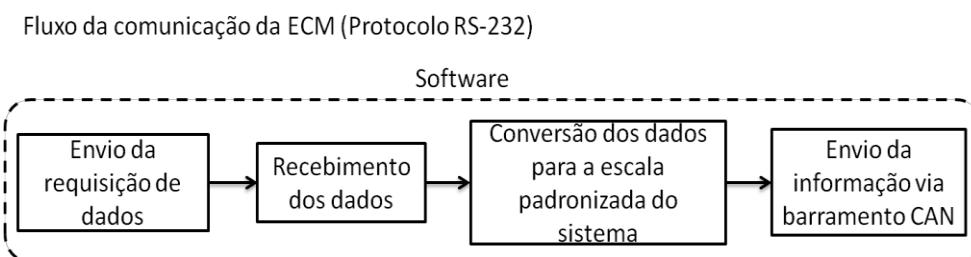


Figura 13 - Fluxo da comunicação da ECM

A Figura 13 mostra a ordem da comunicação com a ECM, unidade que centraliza os dados de temperatura, uptime, tensão de bateria, TPS, RPM, sensor lambda e outros necessários para controle da injeção de combustível para o motor



Figura 14 - Unidade de Controle do Motor (ECM)

Na Figura 14 tem-se a imagem da ECM utilizada pela equipe, cujo desenvolvimento foi feito pela empresa parceira, e algumas modificações e melhorias foram solicitadas pela nossa equipe de desenvolvimento, que podemos citar a falha de comunicação segundo o protocolo CAN.

Fluxo da leitura de um dado digital (borboletas do sistema de troca de marchas *shifter*)

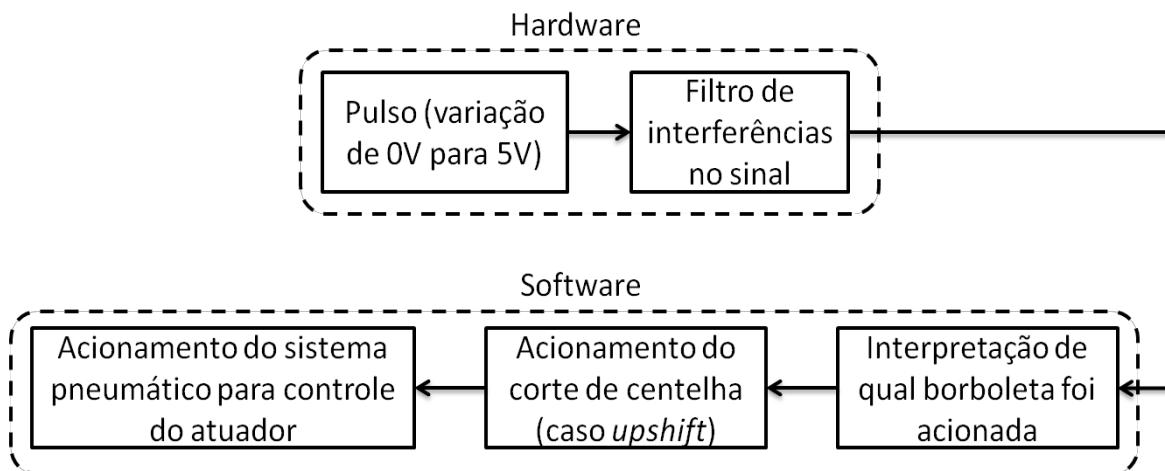
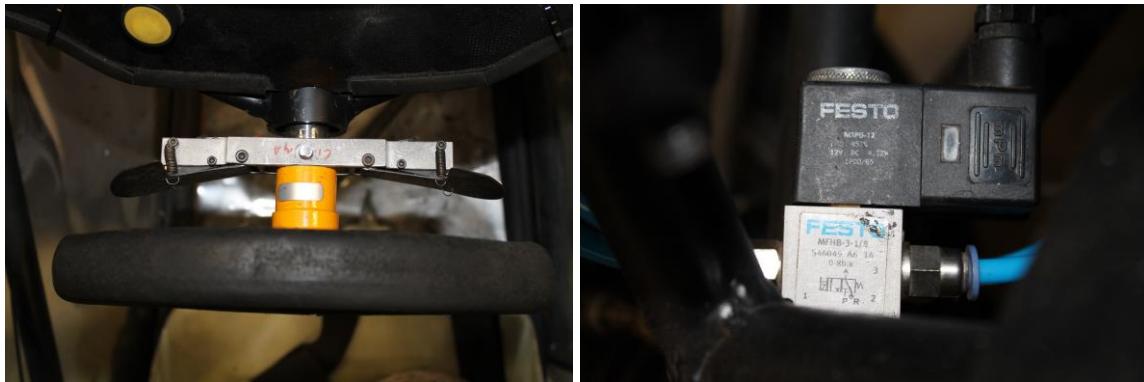


Figura 15 - Fluxo da leitura de um dado digital (Atuadores)

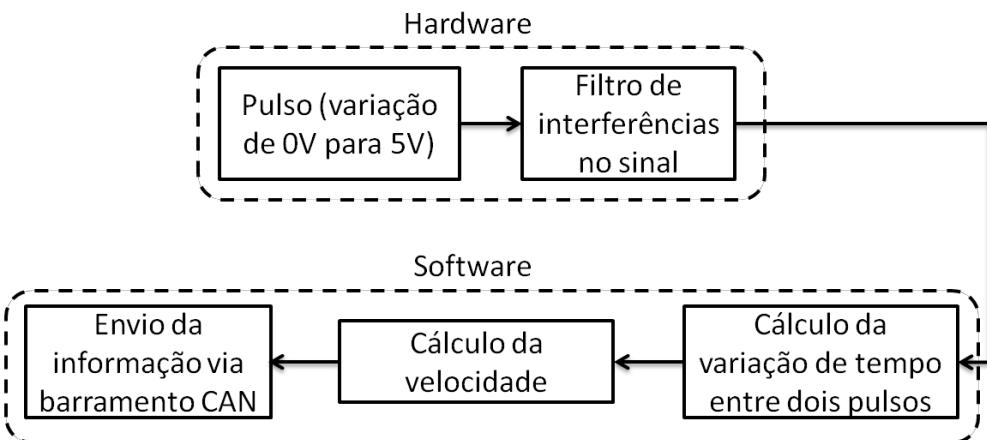
A Figura 15 nos mostra o fluxo do funcionamento do sistema de troca de marchas, desde seu acionamento pelo piloto com uso de borboletas (botão) até o acionamento elétrico do atuador pneumático.



**Figura 16 – Borboletas e válvula solenoide para troca de marcha**

As imagens da Figura 16 mostram o sistema Shifter responsável pela troca automatizada de marchas, em que o piloto aciona as borboletas atrás do voltante sem retirar as mãos do volante, favorecendo o melhor desempenho.

Fluxo da leitura de um dado digital (sensor de velocidade)



**Figura 17 - Fluxo de leitura de um dado digital (Sensores)**

A Figura 17 apresenta o fluxo para leitura de dados digitais para determinação da velocidade do protótipo.



**Figura 18 - Sensor de velocidade**

Na Figura 18 encontra-se o sensor para determinar velocidade, nota-se o desenvolvimento em conjunto com a equipe de suspensão que projetou um suporte para o mesmo ao desenhar as mangas do protótipo.

### **1.4.3 Projeto de estruturas de dados**

A equipe não utiliza estrutura de dados complexas, sendo utilizadas poucas variáveis comuns a todo projeto. Devido ao uso do protocolo de comunicação CAN, os dados são divididos em mensagens as quais possuem uma tabela de identificação que permite a qualquer módulo, ao ler esta mensagem, saber os dados contidos na mesma.

**Tabela 4 - Mensagens**

Nome da mensagem	Número	Dados contidos
MSG_POPA_DATA	31	DATA[0]: Pressão de Freio 3 High Bits
		DATA[1]: Pressão de Freio 3 Low Bits
		DATA[2]: Pressão de Freio 4 High Bits
		DATA[3]: Pressão de Freio 4 Low Bits
		DATA[4]: Acelerometro CM X
		DATA[5]: Acelerometro CM Y
		DATA[6]: Pressão do Óleo High Bits
		DATA[7]: Pressão do Óleo Low Bits
MSG_POPA_DATA_2	33	DATA[0]: Neutro
		DATA[1]: Velocidade roda 3
		DATA[2]: Velocidade roda 4
		DATA[3]: Contador trocas de marcha High Bits
		DATA[4]: Contador trocas de marcha Low Bits
MSG_PROA_DATA_1	67	DATA[0]: Pressão de Freio 1
		DATA[1]: Pressão de Freio 2
		DATA[2]: Curso pedal de freio
		DATA[3]: Curso da direção
		DATA[4]: Curso da embreagem
		DATA[5]: Velocidade 1
		DATA[6]: Velocidade 2
MSG_PROA_ECMDATA1	71	DATA[0]: Uptime High
		DATA[1]: Uptime Low
		DATA[2]: Temperatura ar admissão
		DATA[3]: Temperatura água motor
		DATA[4]: Tensão da Bateria
		DATA[5]: Status sensor de fase
MSG_PROA_ECMDATA2	73	DATA[0]: TPS
		DATA[1]: RPM High
		DATA[2]: RPM Low
		DATA[3]: Lambda High
		DATA[4]: Lambda Low
		DATA[5]: MAP High
		DATA[6]: MAP Low

#### 1.4.4 Projeto de algoritmo

A equipe utiliza o software MPLAB IDE para codificação e gravação, sendo o código escrito em linguagem C, devido ao conhecimento adquirido em diversas disciplinas do curso que facilita sua manipulação, e por permitir escrita, compilação e gravação em

um único ambiente. Durante o desenvolvimento do código existe uma grande preocupação em comentá-lo visando facilitar manutenções futuras.

A Tabela 5 mostra informações sobre norma, resolução e intervalo dos dados módulos responsáveis pela aquisição:

**Tabela 5 - Normas, resoluções e intervalo dos principais dados**

Módulo	Dado	Norma	Resolution	Data range
Proa	Pressão freio 1	SAEpr16	50 kPa/bit	0 to 12,5 MPa
	Pressão freio 2	SAEpr16	50 kPa/bit	0 to 12,5 MPa
	Curso pedal de freio	SAEpc03	0,4 %/bit, 0 offset	0 to 100 %
	Curso direção	SAEpc04	0,8%/bit, 0 offset	-100% to 100%
	Curso embreagem	SAEpc03	0,4 %/bit, 0 offset	0 to 100 %
	Velocidade 1	-	0,2 m/s per bit, 0 m/s offset	0 to 51 m/s
	Velocidade 2	-	0,2 m/s per bit, 0 m/s offset	0 to 51 m/s
Popa	Pressão óleo	SAEpr11	5 kPa/bit	0 to 1,25 MPa
	Pressão freio 3	SAEpr16	50 kPa/bit	0 to 12,5 MPa
	Pressão freio 4	SAEpr16	50 kPa/bit	0 to 12,5 MPa
	Troca de marchas	-	1 troca/bit	0 a 65535 trocas
	Acelerômetro X	SAEac02	0,1 m/s <sup>2</sup> per bit, -12,5 m/s <sup>2</sup> offset	-12.5 to +12.5 m/s <sup>2</sup>
	Acelerômetro Y	SAEac02	0,1 m/s <sup>2</sup> per bit, -12,5 m/s <sup>2</sup> offset	-12.5 to +12.5 m/s <sup>2</sup>
	Velocidade 3	-	0,2 m/s per bit, 0 m/s offset	0 to 51 m/s
	Velocidade 4	-	0,2 m/s per bit, 0 m/s offset	0 to 51 m/s
ECM	Uptime	SAEtm06	1 s/bit	0 to 65535 s
	Temp. ar admissão	SPN 105	1 deg C/bit, -40 deg C offset	-40 to 210 deg C
	Temp. água motor	SPN 110	1 deg C/bit, -40 deg C offset	-40 to 210 deg C
	Tensão bateria	-	0,1V/bit, 0 offset	0 to 25,5V
	Status sensor de fase	SAEr03	1/bit	0 to 255
	TPS	SPN 51	0,4 %/bit, 0 offset	0 to 100 %
	RPM	-	1 rpm/bit	0 to 65535 rpm
	Lambda	SAEr01	0,001/bit	0 to 65,535
	MAP	SAEpr03	0,1 kPa/bit	0 to 6425,5 kPa

Segue fluxograma dos procedimentos realizados para leitura, tratamento e cálculos e disponibilização na rede.



**Figura 19 - Fluxograma procedimentos programação**

## 1.5 Testes

### 1.5.1 Teste de Componente

Descrevemos os testes dos componentes individuais do sistema, expondo os defeitos encontrados.

ECM – falha de comunicação e tratamento de dados.

ECM – não permitia corte de centelha para troca de marchas.

Os outros componentes não apresentaram falhas individuais durante os testes.

### 1.5.2 Teste de Sistema

Teste de comunicação e teste entre os módulos para integração do sistema. Foi encontrada uma falha na comunicação entre os módulos PROA e ECM. Este não respondia à requisição de transmissão de dados que aquele enviaava.

### 1.5.3 Teste de Aceitação

O teste de aceitação foi realizado antes do lançamento do protótipo, realizado no dia 08 de Agosto, atendendo requisitos mínimos de funcionamento, como controle do motor e sensoriamento parcial. Desta forma, as atividades corretivas e adaptativas realizadas a fim de alcançar a especificação inicial encontra-se descrita nas atividades de manutenção.

## **1.6 Manutenção**

Descrevemos aqui as atividades de manutenção realizadas durante a implementação do projeto de software, de forma a contemplar a documentação, auxiliando manutenções e entendimento futuro deste projeto.

Dentro deste projeto Formula SAE, consideraremos manutenção toda atividade realizada após a primeira entrega em funcionamento de cada sistema.

### **1.6.1 Manutenção Corretiva**

Durante implementação do software feito pela equipe, detectou-se problemas neste. A partir disto, foram feitas correções no próprio software a fim de sanar os problemas detectados. Por exemplo, quando foi requerido o envio de um sinal booleano (0 ou 1) dependente da detecção de um valor específico de um sensor.

### **1.6.2 Manutenção Adaptativa**

No módulo PROA havia um C.I. MAX232 para comunicação serial e foi retirado

## **Capítulo 2 – Resultados e Conclusões**

### **2.1 Conclusões Gerais**

A elaboração deste projeto colaborou com o desenvolvimento da documentação e organização do mesmo, assim como permitiu a evolução do grupo de desenvolvedores. Esta documentação contribuirá para futuros trabalhos de manutenção e novos desenvolvimentos.

Aprendemos com a confecção do mesmo que os trabalhos de manutenção e teste podem ser realizados com maior foco, contendo mais informações e descrevendo melhor cada método seguido pelos desenvolvedores independentemente dos resultados alcançados. Tal documentação poderia ser usado como manual de procedimentos para projeto de software.

A execução do projeto atingiu a maior parte dos objetivos, permitindo a evolução dos outros subsistemas com os resultados captados durante os testes do protótipo.