**Demo cluster instrument**

**Customer Rights and Responsibilities**

Bufu Mario

Universitatea Politehnica din Timișoara,

Facultatea de Automatica și Calculatoare

****

Volkswagen full digital cluster instrument

**Table of contents:**

1. Project overview................................................................................................3
2. Project requirements.........................................................................................3
3. Project specifications.........................................................................................3
4. Tasks.................................................................................................................3
5. Gantt chart for the completion of tasks..............................................................4
6. Customer Rights and Responsibilities – an overview........................................4
7. Customer Rights and Responsibilities in the context of the “Demo cluster instrument” project.. ..........................................................................................5
8. Conclusions.......................................................................................................7
9. **Project overview:**

The project’s aim is to design a cluster instrument for presentation that has two operating modes:

* Auto-mode;
* Manual mode (controlled by a pc steering wheel);

1. **Project requirements:**

**Functional requirements**:

* Auto mode must ensure that the cluster instrument will run for many hours without stopping;
* At any point in time when user wants to control the cluster instrument by steering wheel the cluster instrument should switch to manual mode without delay;
* In manual mode the cluster instrument may receive multiple inputs at the same time(ex: right indicator and acceleration pedal pressed);

**Technical requirements**:

* Extra hardware to transfer signals from USB to CAN:
* Implementation of required software to get signals from steering wheel via USB and send them to cluster instrument via CAN;
* Proper documentation for buttons mapping;

1. **Project specifications:**

The project will use the following components:

* Cluster instrument;
* Pc steering wheel;
* Arduino board used for transfer signals from USB to CAN;
* Connection between the steering wheel and cluster instrument will be ensured by Arduino board;
* Power supply;

**4. Tasks:**

1. Project research and product specifications and requirements analysis;
2. Hardware construction of connection between cluster instrument and pc steering wheel;
3. Software development: - demo module for cluster instrument;

-arduino module for transfer signals from USB to CAN ;

1. Software module testing and debugging;
2. Software system testing and debugging;
3. Documenting the product;
4. **Gantt chart for the completion of tasks**
5. **Customer Rights and Responsibilities – an overview**

Software success depends on developing a collaborative partnership between

software developers and their customers.

Problems arise partly because people don't share a clear understanding of what

requirements are and who the customers are.

To clarify key aspects of the customer-developer partnership, Karl Wiegers [\*]

proposes two documents:

(1) Requirements Bill of Rights for software customers

(2) Requirements Bill of Responsibilities for software customers

Because it's impossible to identify every requirement early in a project, the commonly

used ( and sometimes abused) practice of requirements sign-off bears further

examination.

A customer is anyone who derives direct or indirect benefit from a product. This

includes:

(1) People who request, pay for, select, specify or use a software product, or

(2) Those who receive the product's outputs.

There are different types of customers:

(1) Customers who initiate or fund a software project.

o They supply the high-level product concept and the project's business

rationale.

o These business requirements describe the value that the users, developing

organization or other stakeholders want to receive from the system.

(2) Customers who will actually use the product. Users can describe:

o The tasks they need to perform—the use cases—with the product

o The product's desired quality attributes.

o Analysts interact with customers to gather and document requirements and

derive specific software functional requirements from the user requirements.

(3) For commercial software development, the customer and user are often the

same person.

o Even for commercial software you should get actual users involved in the

requirements-gathering process, perhaps through focus groups or by building

on your existing beta testing relationships.

(4) Customer surrogates, such as the marketing department, attempt to determine

what the actual customers will find appealing.

**Customer rights**

Right #1: To expect analysts to speak your language.

Right #2: To expect analysts to learn about your business.

Right #3: To expect analysts to write a Software Requirements Specification(SRS).

Right #4: To have developers explain requirements work products.

Right #5: To expect developers to treat you with respect.

Right #6: To have analysts’ present ideas and alternatives for requirements

and implementation.

Right #7: To describe characteristics that will make the product easy and

enjoyable to use.

Right #8: To be presented with opportunities to adjust your requirements to

permit reuse**.**

Right #9: To be given good-faith estimates of the costs of changes.

Right #10: To receive a system that meets your functional and quality needs.

**Customer responsibilities**

Responsibility #1: To educate analysts about your business.

Responsibility #2: To spend the time to provide and clarify requirements.

Responsibility #3: To be specific and precise about requirements.

Responsibility #4: To make timely decisions.

Responsibility #5: To respect a developer's assessment of cost and feasibility.

Responsibility #6: To set requirement priorities.

Responsibility #7: To review requirements documents and prototypes.

Responsibility #8: To promptly communicate changes to the product's

requirements.

Responsibility #9: To follow the development organization's requirements

change process.

Responsibility #9: To follow the development organization's requirements

change process.

**Customer Rights and Responsibilities in the context of the “Demo cluster instrument” project:**

As work on the project will begin, certain specifications or requirements may need to be modified, and some may even need to be added; Changes in the code will appear, to reflect the new requirements and specifications;

We will aim to update requirements and specifications, to ensure consistency of the product’s performance, functional and physical attributes with its requirements, design and operational information at any given time throughout the project’s life, and to track changes brought to the code. This will be done with the help of documents and a versioning system;

The project will be separated into tasks, and each participant will have tasks assigned to them; Participants will have access to the project Gantt chart, where the new tasks will appear, along with the name of the assigned responsible of the change needed; once the change has been made, tested and integrated the task can be set to DONE;

As a whole, the project will be separated into two components: the code component, and the documents component.

The code component will comprise of:

* Demo module;
* Signals transfer module;

The documents component will comprise of:

* Project requirements document;
* Project specifications document;
* General project documentation of the final product;
* Project test specification- one for each code component;

Changes will be tracked with the help of a versioning system (Git, PTC Integrity); each user will create an account, through which their own modifications can be tracked throughout the whole project development;

When a change of code is done, a series of steps will be followed:

1. the last revision of the project specifications and requirements will be checked;
2. the newly added functionality will be tested;
3. the code change will be checked-into the versioning tool, with a relevant description of the modification brought;
4. the general documentation of the product will be updated if needed;
5. a new test will be added in the project test specification, if possible and needed; this test will describe a way to test the newly added functionality;
6. checked-in code will be integrated in the base project;

Once the final version of the product is done, system testing will begin.

The first test will verify that all the requirements and specifications are consistent with the final product; to achieve this, the respective documents will be checked, and the product will be tested in parallel;

The second test will aim to verify that all the tests added in the projects test specification document pass in the final form of the product;

1. **Conclusions**

In the context of the “Demo cluster instrument” project, the customer needs must be fulfilled (consistency of the product’s functionalities with the requirements and specifications throughout the whole product development).

By dividing the project into tasks and assigning them to project participants, an efficient workflow is achieved. Establishing a set of steps to be followed with each code change reduces the bug risks, and using a versioning tool (ex: git) for the code development and document changes makes it easy to find bug sources, or detect with which change one may have started to appear;

Using documents to track modifications and corresponding tests ensures that when the final product exists, it comes with a set of tests to verify that each functionality works. This makes it almost impossible to forget to test some parts of the code in the final product.