Scrum and Embedded Software Development for the Automotive Industry

Ricardo Y. Takahira[1], Lilian R. Laraia[2], Frederico A. Dias[3], Abraham S. Yu[4], Paulo T. S. Nascimento[4], Alceu S. Camargo Jr.[4]
[1]Magneti Marelli Automotive Systems Ltd., Hortolândia – SP – Brazil
[2]Instituto de Pesquisas Energéticas e Nucleares, University of Sao Paulo, Cotia – SP – Brazil
[3]FEA - University of Sao Paulo, Campinas – SP – Brazil
[4]Business Department, FEA / USP, São Paulo – SP – Brazil

Abstract--**Embedded Software is becoming a key buying criterion for new vehicle customers. Workshops, Conferences and Presentations worldwide are showing that SCRUM methodology is increasing software development speed. So far SCRUM has seen little application in automotive industry where V-model, a more rigid methodology, is dominant. Scientific journals have not been sensitive to the opportunity of speeding up the Embedded Software development by applying SCRUM methodology. The purpose of this study is to provide a first-hand investigation about the factors that may speed up the development and integrate SCRUM with the regular methodology in automotive industry. The challenge is to find the best framework to combine SCRUM and the V-model for Systems Engineering Process. Both should be together applied, as a hybrid methodology, to keep the accepted levels of quality and safety powertrain requirements.**

I. INTRODUCTION

In terms of methodology to carry the process of developing new technology for vehicles, the automotive production chains are still very conservative. The challenges in automotive industries regarding the use of SCRUM methods applied for Electronic Control Unit (ECU) software development, for example, are the requires to fulfill the "traditional" development standards [1]. Some others requirements also represents troubles to speed up the development phase such as mature products of high quality, with long lifetime and a guarantee, requires a high degree of product documentation or must consider that the development team of ECU is part of huge intercompany team that is developing a car [1].

The Agile family of development methodologies was born out of a belief that an approach more grounded in human reality would yield better results. It emphasizes building working software that people can get hands on quickly, versus spending a lot of time writing specifications up front. Agile focuses on small, cross-functional teams empowered to make decisions, versus big hierarchies and compartmentalization by function. Agile also focuses on rapid iteration, with as much customer input along the way as possible. Often, when people learn about Agile, there's a glimmer of recognition – it sounds a lot like back in the start-up days, when we "just did it" [2].

One of the fastest-growing Agile methods is SCRUM. Ken Schwaber and Dr. Jeff Sutherland formalized it over a decade ago, and companies large and small are now using it, including Yahoo!, Microsoft, Google, Lockheed Martin, Motorola, SAP, Cisco, GE and the US Federal Reserve. Many teams using SCRUM report significant improvements, and in some cases complete transformations, in both productivity and enthusiasm. For product developers – many of whom have been burned by the "management fad of the month club" – this is significant. SCRUM is simple, powerful, and rooted in common sense [3].

SCRUM project management is a contemporary approach or methodology for managing software development projects. SCRUM is an iterative, incremental framework. This method structures product development in cycles of work called Sprints, iterations of work which are typically 1-4 weeks in length, and which take place one after the other. The Sprints are of fixed duration – they end on a specific date whether the work has been completed or not, and are never extended. At the beginning of each Sprint, a cross-functional team selects items from a prioritized list of requirements, and commits to complete them by the end of the Sprint; during the Sprint, the deliverable does not change. Each workday, the team gathers briefly to report to each other on progress, and update simple charts that orient them to the work remaining. At the end of the Sprint, the team demonstrates what they have built, and gets feedback which can be incorporated in the next Sprint. SCRUM emphasizes producing working product at the end of the Sprint is really "done"; in the case of software, this means code that is fully tested and potentially shippable [3].

Markus Mueller's practical experiences (KUGLER MAAG CIE GmbH - 2011) [1] in automotive industry, besides having consider the existing standards and requirements, he does not recommend the purely SCRUM approach, but to integrate SCRUM elements into existing and proven development cycles, and to take advantage of both worlds. He recognizes that some SCRUM elements have already proved their worth in automotive for years, such as:

- Incremental development in general to delivery of increments/samples and validation of these at the supplier and the customer side; incorporating the return flow of results into the next increment;
- Rough overall release planning, detailed planning only for the next increment;
- Requirements are not fixed at the start, but developed and clarified during development.

Due to the nature of the final product, many software industries are expected to produce high-quality software through the use of processes. In the last decade, there been a

significant growth in the production of automotive software. It is expected that the industrial markets will require process maturity levels from companies to demonstrate the quality of their software. In many countries there is a vibrant software industry with many Small-to-Medium sized Enterprises (SMEs) [4] [5].

The Agile, Hybrid Assessment Method for the Automotive Industry – AHAA – supports the integration of Agile practices with the more traditional plan-driven practices that are associated with SPI – Software Process Improvement – in safety-critical companies [5].

The Software Process Department at Daimler Chrysler Research and Technology is highly professional at developing embedded software for busses and coaches and supports all business units. In 2001 the investigation on Agile approaches started as a potential new source of ingredients for the improvement toolbox. While a number of IT-projects were successfully adopted and applied Agile methods, business units that produce software to be embedded into vehicles seemed to be more reluctant to give agility a try [6].

The goal of this paper is to conduct an overview analysis about the approach to drive the process of embedded software improvement as a hybrid methodology of traditional to the six SCRUM principles. This assessment method integrates the structured of V-Model with the flexibleness of SCRUM practices.

II. V-MODEL SOFTWARE DEVELOPMENT

The V-Model is Software Development Life Cycle – SDLC – where execution of processes happens in a sequential manner in V-shape. It is also known as Verification and Validation mode (see Fig. 1). V-Model is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase. This is a highly disciplined model and next phase starts only after completion of the previous phase [8].

The V-model design represents a software development process (also applicable to hardware development) which may be considered an extension of the waterfall model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. The horizontal and vertical axes represents time or project completeness (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively [9].

V- Model application is almost same as waterfall model, as both the models are of sequential type. Requirements have to be very clear before the project starts, because it is usually expensive to go back and make changes. This model is used in the development field for automotive, medical and others applications, as it is strictly disciplined domain [8].

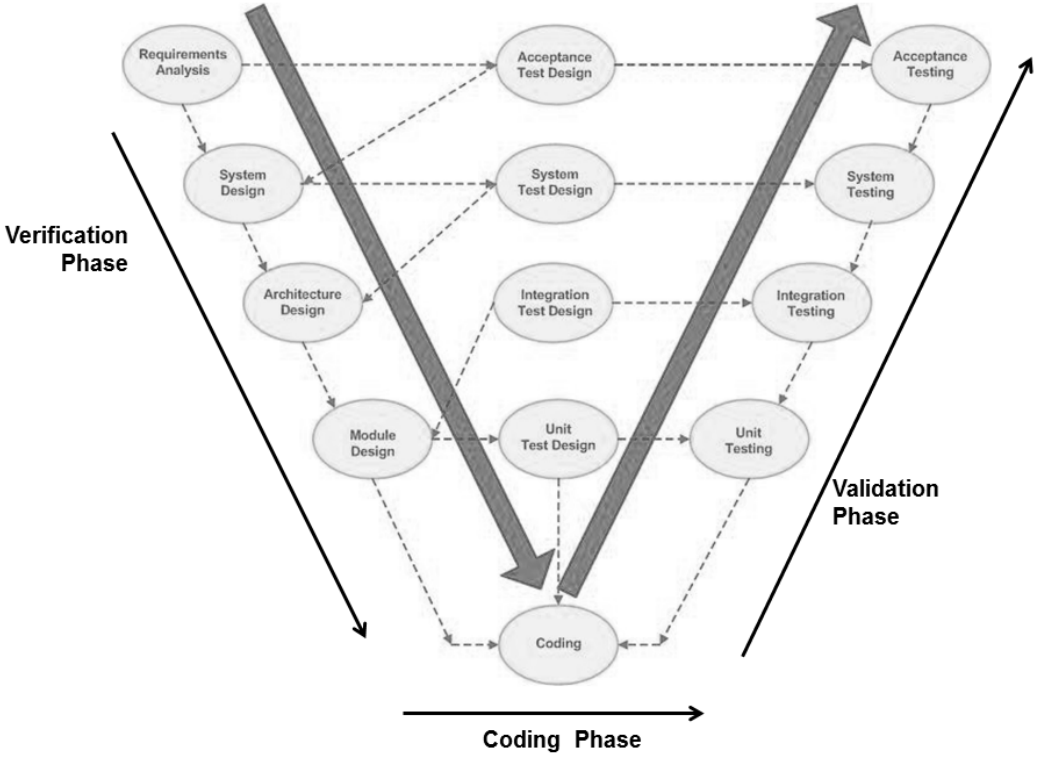

Fig. 1 – Life-cycle of V-Model development process [8] [9]

Following are the suitable scenarios to use V-Model [10]:

- Requirements are well defined, clearly documented and fixed:
  Problem: The internal client (manufacturing) even when participate part time, places no requirements on their experience of the current product. The horizontal dotted line soliciting early involvement rarely happens with efficiency
- Product definition is stable:
  Considering the comment above, when the participation of internal client occurs. If some issue does not meet the requirements and the stage already approved, the methodology is rigid and there is no way to step back to return to previous phase, forcing a poor definition be part of the reality of the project, carrying it to the end.
- Technology is not dynamic and is well understood by the project team:
  One of the factors that decrease the participation and interest of key people and the entire team is a long project lead time, giving the impression that everything is a very distant reality and it is not important to solve or discuss now. Reduce development time and increase the interest and attention of the team is a positive goal for any development activity.
- There are no ambiguous or undefined requirements:
  The lack of interest of the participants conducts to the lack of input requirements
- The project is short:
  It is a, short, desired value with good requirements captured however, due to the sequencing and time involved, it is very difficult to guarantee.

III. SCRUM SOFTWARE DEVELOPMENT

Under the SCRUM approach, the product development process emerges from the constant interaction of a hand-picked, multidisciplinary team whose members work together from start to finish. Rather than moving in defined, highly structured stages, the process is born out of the team members' interplay (see Fig. 2). A group of engineers, for example, may start to design the product (phase three) before all the results of the feasibility tests (phase two) are in. Or, the team may be forced to reconsider a decision as a result of later information. The team does not stop then, but engages in iterative experimentation. This goes on in even the latest phases of the development process [7].

This holistic approach has six principles: built-in instability, self-organizing project teams, overlapping development phases, "multilearning," subtle control, and organizational transfer of learning. The six principles fit together, forming a fast and flexible process for new product development. Just as important, the new approach can act as a change agent: it is a vehicle for introducing creative, market-driven ideas and processes into an old, rigid organization.

Each element, by itself, does not bring about speed and flexibility. But taken as a whole, the characteristics can produce a powerful new set of dynamics that will make a difference [7].

1- Built-in instability: Top management kicks off the development process by signaling a broad goal or a general strategic direction.

2- Self-organizing project teams: A project team takes on a self-organizing character. Ambiguity and fluctuation abound in this state . The most important features are:
Autonomy: the team is free to set its own direction.
Self-transcendence: The project teams appear to be absorbed in a never-ending quest for "the limit".
Cross-fertilization: Team members with varying functional specializations, thought processes, and behavior patterns carries out new product development.

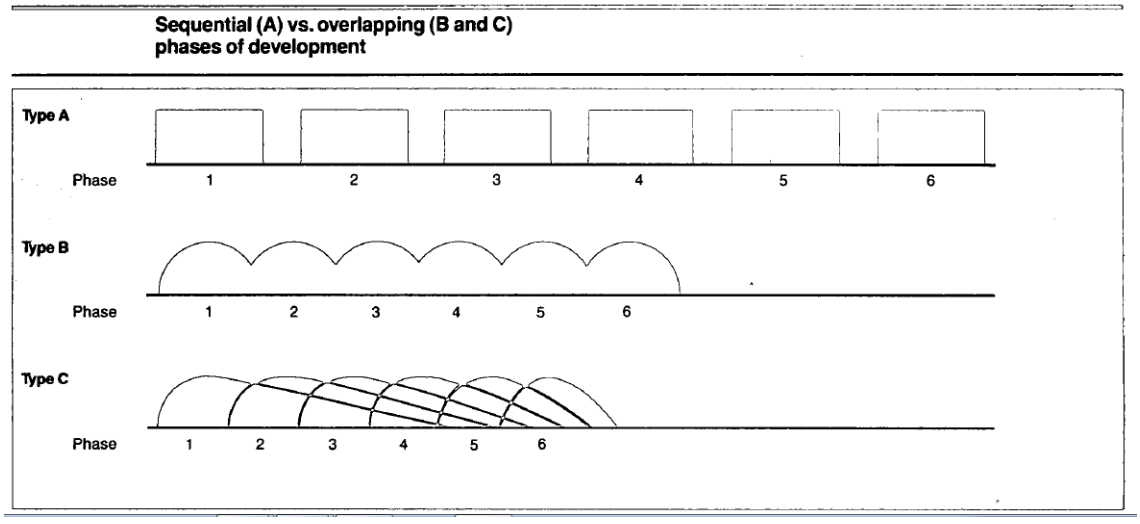

Fig. 2 – New Product Development Process Phases [7]

3- Overlapping development phases: The overlapping approach has greater speed and increased flexibility, which are the "hard" merits (see Fig. 2).
4- Multilearning: Team engage in a continual process of trial and error to narrow down the number of alternatives that they must consider.
5- Subtle control: Team is not uncontrolled. Management establishes enough checkpoints to prevent instability, ambiguity, and tension from turning into chaos.
   Subtle control is exercised in the new product development process in seven ways:
   a. Selecting the right people for the project team: The participation of experts, can indicate the omission of some area on sprint meetings, for example the representative of manufacturing may not indicate your requirement for assembly without screwdriver. If the participant from this area at the meeting does not manifest the specialist should draw attention to the requirement for the overall experience in the processes.
   b. Creating an open work environment: Bring people not focused at the actual moment lived by the process may distract the group, the leader must avoid these distractions and stay focused on the entire team.
   c. Encouraging engineers to go out into the field and listen to what customers and dealers have to say: Make sure that requisites reflect the wishes of the customer or what is more important: actually meets the expectations of the target audience and are not "stuffed" by personal desires.
   d. Establishing an evaluation and reward system based on group performance: Do not create too much criticism, but reveal the really important comments. Least appears more to contribute in terms of lessons learned.
   e. Managing the differences in rhythm throughout the development process: By rapid cycles, keeping the team focused and quickly able to correct the faults or failures regarding requirements and considerations
   f. Tolerating and anticipating mistakes: Use lessons learned and avoid the same mistakes (record existence of Lessons Learning in database).
   g. Encouraging suppliers to become self-organizing: Involving them early during design is a step in the right direction. But the project team should refrain from telling suppliers what to do: Shorter times, greater participation, expert experience can cover and require pointing of his representatives for all specific issues of involved area.
6- Transfer of learning: The drive to accumulate knowledge across levels and functions is only one aspect of learning. The equally strong drive on the part of the project members is to transfer their learning to others outside the group.

SCRUM BASIS

SCRUM is an iterative, incremental framework. SCRUM structures product development in cycles of work called Sprints, iterations of work which are typically 1-4 weeks in length, and which take place one after the other. The Sprints are of fixed duration – they end on a specific date whether the work has been completed or not, and are never extended. At the beginning of each Sprint, a cross-functional team selects items from a prioritized list of requirements, and commits to complete them by the end of the Sprint; during the Sprint, the deliverable does not change. Each work day, the team gathers briefly to report to each other on progress, and update simple charts that orient them to the work remaining. At the end of the Sprint, the team demonstrates what they have built, and gets feedback which can then be incorporated in the next Sprint. SCRUM emphasizes producing working product at the end of the Sprint is really “done”; in the case of software, this means code that is fully tested and potentially shippable (see Fig. 3) [2].

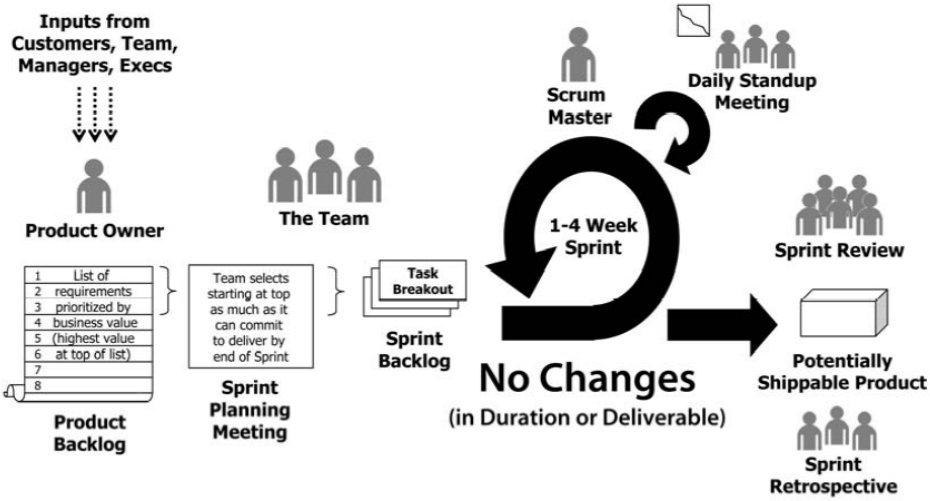

Fig. 3 – SCRUM Software Development Process Phases [2]

SCRUM Roles: In SCRUM, there are three primary roles:

The Product Owner is responsible for achieving maximum business value, by taking all the inputs into what should be produced – from the customer or end-user of the product, as well as from Team Members and Stakeholders – and translating this into a prioritized list. The Product Owner role maps to the Product Manager or Product Marketing Manager position in many organizations.

The Team builds the product that the customer is going to consume: the software or website, for example. The team in SCRUM is "cross-functional" – it includes all the expertise necessary to deliver the potentially shippable product each Sprint – and it is "self-managing", with a very high degree of autonomy and accountability. It's much more productive to have team members fully dedicated. Team members can also change from one Sprint to the next, but that also reduces the productivity of the team. Projects with larger teams are organized as multiple SCRUMs, each focused on a different aspect of the product development, with close coordination of their efforts.

The SCRUM Master is one of the most important elements of SCRUM success. The SCRUM Master does whatever is in their power to help the team be successful. The SCRUM Master is not the manager of the team; instead, the SCRUM Master serves the team, protects the team from outside interference, and guides the team's use of SCRUM.

The Managers are the most important roles. While their role evolves in SCRUM, they remain critically important – they support the team by respecting the rules and spirit of SCRUM, they help remove impediments that the team identifies, and they make their expertise and experience available to the team. In SCRUM, these individuals replace the time they previously spent "playing nanny" (assigning tasks, getting status reports, and other forms of micromanagement) with more time "playing guru" (mentoring, coaching, helping remove obstacles, helping problem-solve, providing creative input, and guiding the skills development of team members) [2].

One of the key pillars of SCRUM is that once the Team makes its commitment, any changes from the Product Owner must be deferred until the next Sprint. This means that if halfway through the Sprint the Product Owner decides that there is a new item they'd like the team to work on, they cannot make the change until the start of the next Sprint.

Once the Sprint has started, the Team engages in another of the key SCRUM practices: The Daily SCRUM. This is a short (15 minutes or less) stand-up meeting that happens every workday at an appointed time, and everyone on the SCRUM Team attends; in order to keep it brief, everyone stands (hence "stand-up meeting"). It's the team's opportunity to report to itself on progress and obstacles.

After the Sprint ends, there is the Sprint Review, where the team demos what they've built during the Sprint. Present at this meeting are the Product Owner, Team Members, and SCRUM Master, plus customers, stakeholders, experts, executives, and anyone else interested. This is not a "presentation" the team gives – there are no PowerPoints, and typically no more than 30 minutes is spent preparing for it – it's literally just a demo of what's been built, and anyone present is free to ask questions and give input. It can last 10 minutes, or it can last two hours – whatever it takes to show what's been built and get feedback [2].

It's an opportunity for the team to discuss what's working and what's not working, and agree on changes to try. The SCRUM Team, the Product Owner, and the SCRUM Master will all attend, and a neutral outsider will facilitate the meeting; a good approach is for SCRUM Masters to facilitate each others' retrospectives, which enables cross-pollination among teams.

A simple way to structure the Sprint Retrospective is to draw two columns on a whiteboard, labeled "What's Working Well" and "What Could Work Better" – and then go around the room, with each person adding one or more items to either list. As items are repeated, checkmarks are added next to them, so the common items become clear.

Following the Sprint Review Meeting, the Product Owner takes all the input, as well as all new priorities that have appeared during the Sprint, and incorporates them into the Product Backlog; new items are added, and existing ones are modified, reordered, or deleted. Once this updating of the Product Backlog is complete, the cycle is ready to begin all over again, with the next Sprint Planning Meeting.

Sprints continue until the Product Owner decides the product is almost ready for release, at which point there may be a "Release Sprint" to do final integration and testing in preparation for launch. If the team has followed good development practices along the way, with continuous refactoring and integration, and effective testing during each Sprint, there should be little pre-release stabilization required [2].

IV. SCRUM/V-MODEL HYBRID METHODOLOGY SOFTWARE DEVELOPMENT

This study purpose an investigation about the condition that may compose the best framework to combine SCRUM and the V-Model. Both methodologies could be together applied as a hybrid methodology for Automotive Software Development. The focus is to speed up the development phase and integrate the best principles and successful practices of SCRUM inside the V-Model methodology to keep the accepted levels of quality and safety powertrain requirements.

The research project was carried out considering the exploratory type, because the main goal is to provide greater conection with the issue. It was divided in two phases: the first phase was to make a breakdown analysis for both methodologies based on the papers review and the second phase was done by discussing with Brazilians specialists on software development from automotive industry. This research proposed a hybrid methodology and framework by doing an analysis considering the relationships of variables for both software development processes and the integration of the most important phases, looking for the development of

embedded software speed up and preserving the requirements for quality and safety..

Phase 1- Breakdown of SCRUM and V-Model Script Framework

The Table 1 reported the analysis of both methodologies considering the most important factors to design a new framework for SCRUM and the V-Model as a hybrid methodology. During the papers review it was possible to identify that the most important factors was: Structure, process, roles and principles. The table 1 present the framework for the SCRUM followed by the V-model, both will guide the new proposed methodology.

The Plan of Projects activities is based on short cycles of activities (Sprints). The duration of each sprint will be no longer than 4 weeks. In-between sprints it will be possible to make a review and consider the activities for the next sprint (interactions). For each sprints it is defined what are the deliverables. The backlog of product and backlog of the activities related to the step of project according to the V-Model.

The proposed Comparison Chart for the current and proposed methodology is divided in three important steps. The first step is to prepare the Product Backlog: list of requirements in priority order sorting them in order of value for the client and for business. The first item is the highest priority. At the second step it is possible to prepare the Sprint Planning: meeting with the responsible to carefully prepare the project planning. The planned dates are considered, include the absences from the office, meetings and other. The Backlog of the sprint will follow the steps of V-Model according to each phase (Requirement Analyses, Coding and Validation). At the third step the Daily SCRUM Process is planned: to update the sprint, to make the sprint review, retrospective, released and the upcoming new sprint.

This study had also considered the participants, team, roles and features. The six principles also represent the most important issue because they are responsible for the most significant contribution for the proposed methodology,

Phase 2- Meetings with the Brazilians specialists on software development

The match was done by individual meeting with eight software development experts from auto-parts industries. They are development engineers working on software development for more than 10 years at 10 different companies. Following are the results:

1 They have never tried the SCRUM methodologies for software development before, but they would like to take an opportunity to try.
2 They realized that the number of Electronic Control Units (ECUs) is increasing – consequently, more interfaces need to be developed between these ECUs.
3 While much automotive software is not safety-critical, for example, software within navigation and infotainment systems, safety-critical functions, such as braking, are becoming more software dependents.
4 While the driver can intervene in the execution of these functions, the increase in software dependence requires that operational safety must be accounted for when developing automotive software.
5 There is also a need for support processes such as configuration management, project management (including risk management), requirements management, subcontractor management and quality assurance.
6 Due to the challenges of controlling the increased complexity that innovations bring, a better process for development software should be developed to establish a common approach for determining software capability/maturity.
7 There is an increasing requirement for the quality of software to be monitored and assessed throughout the development process.

Besides the matching with specialists on software development's results do not represent a positive or negative answer for Agile methodologies application, they presented a big demand and acceptance from the users to improve the current and conservative methodology and keep the need for safety and quality functions.

Based on the experts' matching and on the SCRUM and V-Model methodologies analysis, it was possible to propose a new methodology to carry the Embedded Software Development.

The first step in the proposed Framework for Hybrid Methodology the Product Owner with the customer requirement will prepare product vision. This takes the form of a prioritized list of what's required, ranked in order of value to the customer and business. This is called the Product Backlog, and it exists (and evolves) over the lifetime of the project. It will include a variety of items, such as features, development requirements, exploratory work and known bugs.

The second phase of the process is devoted to the composition of the project team and the planning meeting of the phases and activities of the project. The Project Team is multidisciplinary, comprising professionals who represent the skills needed to perform the three phases of the V-Model: verification, coding and validation. The team is lean and responsible for all project activities, they participate together for all tasks and decisions. They carry out one project at a time, work focused and targeted on meeting the scope set initially and have the autonomy to make decisions and plan the activities of each Sprint. So the professional responsible for Acceptance Test phase (last step) will be involved directly in daily meetings on all earlier phases. By this way it is possible to predict future problems in the next stages and even anticipate effective and/or preventive actions. The internal dynamics results of the group will promote greater use of the knowledge and skills and reduce design time and anticipate future activities.

TABLE 1 – FRAMEWORK COMPARISON CHART FOR METHODOLOGIES

FRAMEWORK DESCRIPTION		FEATURES	SCRUM	V- MODEL	PROPOSED HYBID METHODOLOGY
Structure	Plan Activities	. In cycles of work - Sprints	✓	✗	✓
		. 1-4 weeks in length, one after the other	✓	✗	✓
		. Iterations of work	✓	✗	✓
		. fixed Sprints duration and end	✓	✗	✓
		. deliverable does not change.	✓	✗	✓
		. Daily report, update Charts to each other on progress	✓	✗	✓
		. Sprint's finished with the deliverables and feedbacks	✓	✓	✓
		. incorporations for next Sprint	✓	✓	✓
		. whole project's activities planed at the beginning	!	✓	!
		. Whole requirements are defined by the project's start	!	✓	!
		. highly disciplined model and Phases no change for whole phases	✗	✓	✗
		. new activities can be added	✓	✗	✓
Process	Steps	**1st. Step**	Prepare the Product Backlog	Requirements Analysis	Prepare Product Backlog for Requirements Analysis, Coding and Validation
		2nd. Step	Prepare the Sprint Planning:	Coding	Prepare Sprint Planning for Requirements Analysis, Coding and Validation
		3rd. Step	Daily Scrum Process	Validation	Daily Scrum Process: update the sprint, make the sprint Review, Retrospective, Released and the upcoming new sprint
Roles	Product Owner	Responsible for achieving maximum business value and translating this into a prioritized list	✓	✗	✓
	Manager	Represent the client expectation and no interaction with the team	They play micromanagement and replace the time if necessary	Represent the client expectation	They play micromanagement and replace the time if necessary
	Team	One by cross-function team of to builds the whole product	✓	✗	✓
		One team of experts for each step of process / divided in areas	✗	✓	✗
Principles	1. Built-in instability	. Kicks off signaling a broad project goal	✓	✗	✓
		. Team is challenged for only one phase of the whole project	✗	✓	✗
		. General strategic direction for whole team	✓	✗	✓
		. Each department knows what to do	✗	✓	✗
		. General strategic direction for whole team	✓	✗	✓
		. Oferred to the Project team a wide measure of freedom	✓	✗	✓
		. Establishes extremely challenging goals	✓	✗	✓
		. Top manager creates an chalenger at the project team	✓	✗	✓
		. great freedom to carry out a project of strategic importance	✓	✗	✓
		. Top manager sett very challenging requirements.	✓	✗	✓
	2. Self-organizing project teams	. Autonomy: team takes on a self-organizing character and is free to set its own direction.	✓	✗	✓
		. Self-transcendence: teams is absorbed in na ever-ending quest for "the limit."	✓	✗	✓
		. Cross-Fertilization: team consisting of members with varying functional specializations, thought processes and behavior patterns carries out new product development.	✓	✗	✓
	3. Overlapping development phases	The individual's rhythm and the group's rhythm begin to overlap, creating a whole new pulse. This pulse serves as the driving force and moves the team forward.	✓	✗	✓
	4. Multilearning	They acquire broad knowledge and diverse skills, which help them create a versatile team capable of solving an array of problems fast.	✓	✗	✓
	5. Subtle control	Management establishes enough checkpoints to prevent instability, ambiguity, and tension from turning into chaos	✓	✗	✓
	6. Transfer of learning	Transfer of learning to subsequent new product development projects or to other divisions	✓	✗	✓

Reading:

✓ Yes

! Partial

✗ No

For the traditional methodology the steps of phases are performed by specialists and each one is further step occurs regardless and in different departments with hierarchical structure. There is no communication, responsibilities or exchange detailed information between the phases (departments).

The third and final phase was developed with Sprint rounds for each step of the process. The project team itself will define all deliveries, responsible activities and deadlines. At the end of each round, which can last from 1 to 4 weeks, Sprint is closed and the project team assesses the results and the activities carried out, if there are any pending or activity that needs to be repeated, this may be included in the next Sprint.

What actually occurs is an overlapping between the stages of the project, because they may be anticipated or pending activities can be performed in the next stage, since planned for the team and considered in the Backlog for the next Sprint. See the three phases and project development on Fig.7.

V. CONCLUSION

This paper presents the first investigation about how HYBRID METHODOLOGIE PROCESS IMPROVEMENT WITH SCRUM AND V-MODEL was conducted in a safety critical software development environment, based on the extensive authors' hands-on experience applying these methods, informal interviews with experienced professionals on embedded software for automotive industry, as well as reviewing the literature and previously published articles.

The study brings forward a hybrid conceptual model proposal, combining. A hybrid model was developed in order to keep the best side of SCRUM and V-Model can be done together. V-Model phase structure with SCRUM flexibility, disciplined phase transitions and project management methodology, to increase deployment speed, ensuring quality, specifications fulfillment and safety requirements.

Considering the current scenario where new technology emerges and becomes obsolete with great speed, the authors' conclusion that is entirely possible and highly recommended combining the virtues of these two development methods to overcome the weaknesses that they have applied alone.

This paper was also able to conduct an overview analysis about the approach to drive the process of embedded software improvement as a hybrid methodology of traditional to the six SCRUM principles. This assessment method integrates the structured of V-Model with the flexibleness of Agile practices.

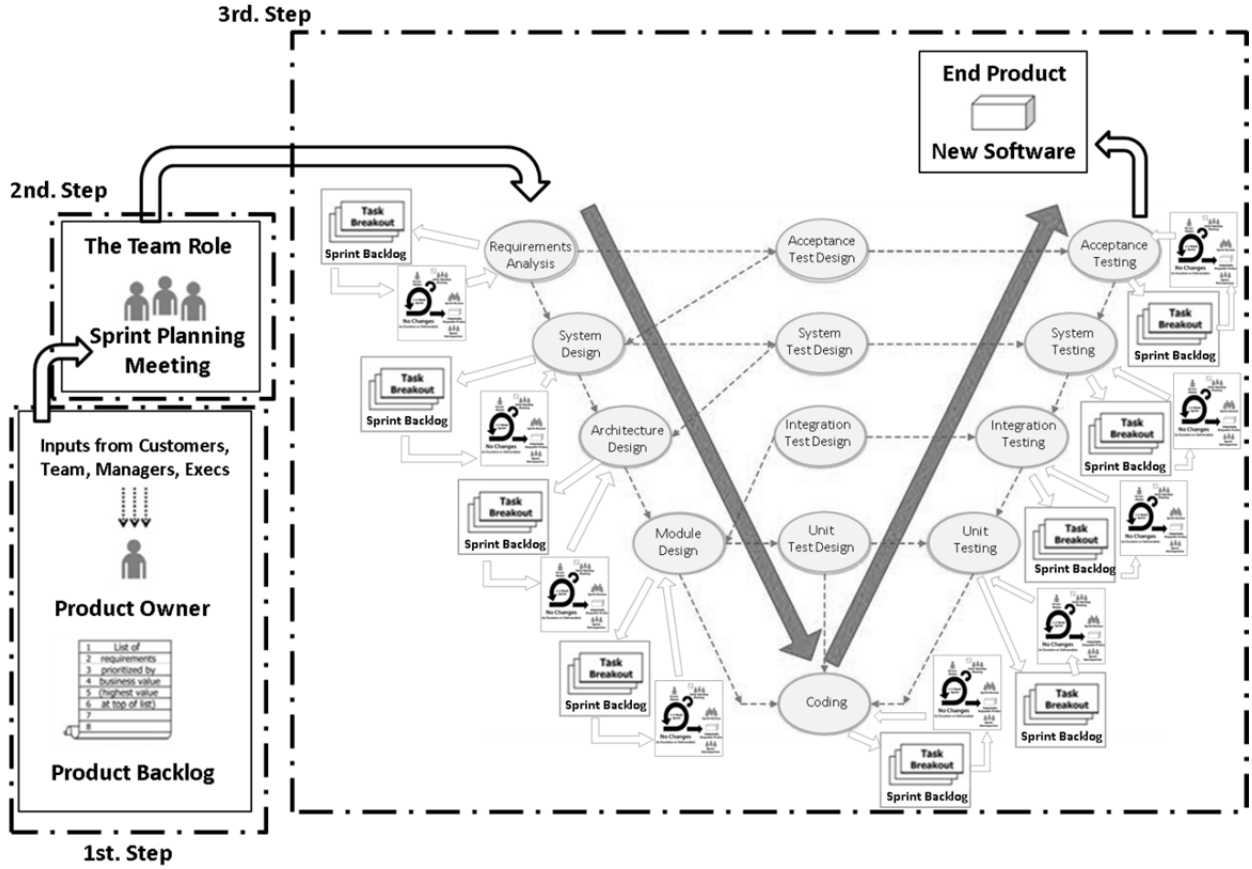

Fig. 7 – Proposed Framework for Hybrid Methodology Process for Embedded Software Development (Figure adapted by the authors from [2] and [9])

Predicting future failures in the implementation of the new methodology:

- This methodology cannot be applied in any kind of development; if the product does not have favorable characteristics will be difficulty and even a failure;
- The experience of team or reluctance to adopt the new methodology may not generate the expected results
- Like any other model, the adoption of new Agile methodology can also fail to be implemented on essence, due usual methodology issues: Members Participations, lack of participants in specific basic roles.
- Lack of Seniority in the chosen experts, absence of identification by the SCRUM Master of points not covered activities.

VI. LIMITATIONS OF THESE STUDY AND FUTURE WORK

The scope of this research has some limitation despite achieving the proposed objective:

- The process development was limited on the extensive authors' hands-on experience, informal interviews with and reviewing the literature and previously published articles
- At the time, it was not able to carry the experimental try or other way to check and adjust the proposed methodology on real product development project routine.
- Due to safety issues, automotive development processes are so rigid and people are very resistant to changes.
- People are not used to SCRUM proposed teamwork, because V-Model phases in fact represent different departments.
- Considering multidisciplinary team working together from start to finish as the primary success factor, relational skills must be evaluated besides technical capacity on the team building.
- And, just as important as the team is the top management willingness to grant great freedom to carry out a project of strategic importance to this.

However, as new approaches to product development gain acceptance in the United States, it is expected that having a persistent practice and continuous improvement for the new methodology, real values with lower investments should speed up the software development for automotive industry with necessary quality and safety.

REFERENCES

[1] M. Markus, "Functional Safety, Automotive SPICE® and Agile Methodology Automotive SPIN", Italia, The Automotive Software Workshop, 17 February 2011.

[2] B. Gabrielle and D. Pete - "The SCRUM Primer – An Introduction to Agile Project Management with SCRUM", © 2007.

[3] S. P. Sone, "Mapping Agile Project Management Practices to Project Management Challenges For Software Development".

[4] O. Salo and P. Abrahamsson, Agile methods in European embedded software development organizations: a survey on the actual use and usefulness of Extreme Programming and SCRUM.

[5] C. Fergal Mc, P. Minna and R. Ita, "AHAA –Agile, Hybrid Assessment Method for Automotive, Safety Critical SMEs".

[6] M. Peter and S. Kurt, "Breaking the Ice for Agile Development of Embedded Software: An Industry Experience Report" © 2004 IEEE.

[7] T. Hirotaka and N. Ikujiro, "The new new product development game Stop running the relay race and take up rugby - High-Speed Management for the High-Itch Age", Fortune,March 5,1984, p. 38.

[8] "Software Development Life Cycle (SDLC)", Tutorials Point, Simply Easy Learning.

[9] "V-Model Software Development", From Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/V-Model_(software_development).

[10] "The Death of the V-Model", From Harmonic Software System, http://www.harmonicss.co.uk/index.php/tutorials/software-engineering/56-the-death-of-the-v-model