Solutions to Automotive Software Engineering Challenges

Adnan Shaout and Gamal Waza
The Electrical and Computer Engineering Department
The College of Engineering and Computer Science
The University of Michigan - Dearborn
Dearborn, MI

Abstract - *The amount of software in cars has been growing over the years. Furthermore, cars in the future also expect to demand more and more software functionalities. Adding to the complexity, the automotive industry has specific constraints and requirements that mandate unique solutions. These specific constraints also bring many challenges to software engineering in cars. In this paper, the automotive software engineering concerns and challenges will be addressed. Software engineering methods such as traditional, agile and/or spiral will be evaluated against the unique environment found in the automotive industry. The Decision Analysis and Resolution (DAR) evaluation process was chosen for evaluating the seven most used software engineering methods in automotive industry with respect to selected prioritized parameters. The selected parameters for the evaluation were Reliability, Low Cost, Reusability, Flexibility, Adaptability, Sustainability, Certifiable, Suitability, Scalability, Simplicity, Efficiency and Controllability. Results have shown that the Spiral software development model is the most suitable process for automotive software engineering. Spiral model scored didn't score high in all parameters. For example, "low cost" parameter was one of the parameters where spiral model scored low. Finally, although evaluation shows spiral was the most suited based on the prioritized parameters, slight change in parameter priority can results in different software development process.*

Keywords **-** *Software Engineering Methods; Automotive; Software Performance Parameters; DAR Process; Software Challenges*

1. INTRODUCTION

In the past 30 years, the amount of software in cars has been growing. Cars in the future also expect to demand more and more software functionalities [14]. In addition, the automotive industry has specific constraints and requirements that mandate unique solutions. These specific constraints also bring many challenges to software engineering in cars. Therefore, there is an opportunity for an automotive software engineering research to help solve many current and future challenges [1].
Automotive software engineering current and future challenges threaten cars software development cost, quality and time-to-market. Below are some current and future challenges and concerns [1][14]:

A. Key Software Development Concerns and Challenges
   a. Demand for lower Cost
   b. Demand for quick delivery of software
   c. Demand for higher quality and reliability
B. Architecture Concerns and Challenges:
   a. Quickly changing platforms and system infrastructure
   b. Reuse of software from car to the next is not sufficient. This is due to the hardware specific optimization that causes 90% of software to be rewritten,
   c. There are many dependencies among the different functions in cars. This makes it necessary to understand cars as a complex system. So, software engineering needs to make a very big step to improve system engineering.
   d. High demand for innovative and improved functionality
C. Others:
   a. Complex requirements

This paper is organized as follows: section 2 will present seven most used software engineering methods as a solution for automotive software engineering challenges. The methods will be evaluated using the Decision Analysis and Resolution (DAR) evaluation process with respect to prioritized parameters that are used in automotive industries. Section 3 will discuss the results of the DAR evaluations. Section 4 will present few conclusion remarks and finally section 5 will present future research.

2. SOFTWARE ENGINEERING METHODS AS A SOLUTION

Software engineering methods helps achieve delivering high quality software on time and within allocated budget [11]. One of the key software engineering methods is to have a suitable Software Development Process (SDP). Selecting the appropriate SDP for the right domain is vital to software development cost, quality and delivery. Many Software Development Processes are evaluated against prioritized list of parameters for the automotive industry domain.

2.1. Selected Software Developments Processes

The following is a list of the most used software engineering methods in automotive industry:

2.1.1. V-Model
2.1.2. Spiral
2.1.3. Rapid Application Development
2.1.4. Iterative Development Process
2.1.5. XP
2.1.6. SCRUM
2.1.7. RUP

2.2. Assumptions

The following are some of the assumptions that the automotive industry has with respect to software product:

2.2.1 Requirements is not stable
2.2.2 There are safety related features
2.2.3 Moderate demand for low cost as compared to reliability and adaptability
2.2.4 Project needs to be delivered iteratively and incrementally
2.2.5 There is a need to achieve CMMI Level 3
2.2.6 Project is not very complex
2.2.7 No new technology
2.2.8 There are several past similar projects
2.2.9 Large project, but many small teams are formed to efficiently handle the many similar projects

2.3. Evaluation Method

The Decision Analysis and Resolution (DAR) evaluation process [3] was chosen for evaluating the seven most used software engineering methods in automotive industry given in section 2.1 with respect to selected criteria.

2.3.1. DAR Process
The following are the DAR process steps:
2.3.1.1. Create evaluation criteria for evaluating selected SDPs
2.3.1.2. Weigh the importance of each criteria
2.3.1.3. Rate each of the SDPs against each criteria
2.3.1.4. Calculate the effective score using Multiply Attribute Utility Tool (MAUT) which is a linear addition of weight multiplied by scores.

2.3.2. DAR Process Steps Details
This section will present the DAR process steps in more details.
2.3.2.1. Creating evaluation parameters
2.3.2.1.1. First step in the evaluation process was to select parameters for evaluation. Selecting the appropriate list of criteria's was the most challenging task of this research paper. There are many parameters that need to be considered for evaluation. Parameters need to have the following characteristics [5]:

- Must be simple and easy to measure
- General
- Balanced
- Quantitive
- Minimum overlap

The high level parameters are created first and are shown in table 1. Then, detailed and specific sub-parameters are created for each high level one [2][4][5][7][8][10].

Table. 1: A list of high level parameters

High Level Parameters (Objective)
Reliability
Low Cost
Reusability
Flexibility
Adaptability
Sustainability
Certifiable
Suitability
Scalability
Simplicity
Efficiency
Controllability

2.3.2.2. Second step is to weigh the importance of each parameter

Values will be interpreted in this paper as follows:

- 100% means the most important criteria.
- 50% means that the most important criteria (100%) is twice more important than 50% rated criteria.
- 0% means not important at all and will not impact the evaluation.

Table 2 shows the high level criteria's weight of importance that will be used in the performance evaluation in this paper.

Table 2 shows the high level criteria's weight of importance.

High Level Parameters	Weight of Importance
Reliability	100.00%
Low Cost	30.00%
Reusability	30.00%
Flexibility	10.00%
Adaptability	50.00%
Sustainability	20.00%
Certifiable	30.00%
Suitability	80.00%
Scalability	10.00%
Simplicity	20.00%
Efficiency	30.00%
Controllability	10.00%

Next, the importance of each sub-parameter is weighed with respect to the assumption stated in earlier. The high level parameters needed to be broken into more specific and measureable sub-parameters. For example, efficiency is a high level parameter and "emphasis on team work" and "less documentation" are the sub-parameters in the efficiency category. Items under efficiency will be weighed against each other as show in table 3.

Table 3. The Efficiency parameter and its sub-parameters.

Objective	Category	Evaluation Parameter	Weight of Importance wrt same category
Efficiency	Processes	Suitability for projects where processes are documented to maximize productivity and efficiency	100.00%
Efficiency	Team work	Suitability for projects with emphasis on Team work to efficiently complete tasks.	100.00%
Efficiency	Minimum Documentation	Suitability for projects with emphasis on less documentation to efficiently complete tasks.	20.00%
Efficiency	Minimum Overhead	Suitability for projects with emphasis on the product (not process) to efficiently complete tasks.	40.00%

Finally, the effective weight of importance for each sub-parameter is normalized for the high level parameters. In addition, the weight of importance for sub-parameters is also normalized within each category separately. This is important since we want to have control over weight of importance for the high level parameters. The final step is multiply normalized sub parameter weight of importance by its high-level parameter weight of importance. For example, if efficiency is 30% important and its sub-parameter "less documentation" is 50% important within efficiency, then, the effective importance of less documentation is 15% (0.50*0.30=0.15).

2.3.2.3. Rate each of the SDPs against each criteria [11][12][13].

Table 4 shows an example of rating the V-Model against the Spiral model with respect to the criteria of Adaptability.

Table 4. Adaptability rate for the V-Model and Spiral Model.

Objective	Category	Evaluation Parameter	V-Model	Spiral
Adaptability	Dynamism	Suitability for projects with **frequent** Requirements changes	50	90
Adaptability	Dynamism	Suitability for projects where requirements are stable	100	90

2.3.2.4. Calculate the effective score using Multiply Attribute Utility Tool (MAUT) [3].

Table 5 shows the effective score calculations using MAUT for the seven most used software processes in automotive industries with the evaluation criteria selected in table 1.

Table 5. Effective score using MAUT for the software processes and the criteria

		80.55	86.76	71.38	82.77	78.59	73.20	83.02
Evaluation Criteria (Objective)	Importance	V-Model-XT	Spiral	RAD	Iterative Dev. Process	Scrum	XP	RUP
Reliability	23.81%	23.81	23.81	16.67	21.43	21.43	21.43	21.43
Low Cost	7.14%	4.62	4.62	5.22	4.62	6.32	6.32	4.67
Reusability	7.14%	5.26	6.59	6.33	5.68	6.17	6.01	6.27
Flexibility	2.38%	1.19	1.43	1.67	1.67	2.14	2.38	1.90
Adaptability	11.90%	6.49	10.71	10.50	10.50	11.47	11.36	11.47
Sustainability	4.76%	3.88	3.84	2.43	3.88	2.96	2.50	3.19
Certifiable	7.14%	7.14	7.14	5.00	7.14	2.14	1.43	6.43
Suitability	19.05%	14.46	14.89	11.48	14.59	13.89	11.00	15.38
Scalability	2.38%	2.14	2.14	1.19	2.14	0.95	0.48	2.38
Simplicity	4.76%	4.29	4.29	4.29	4.29	3.81	3.33	2.38
Efficiency	7.14%	4.89	5.16	4.95	5.16	5.88	5.77	5.85
Controllability	2.38%	2.38	2.14	1.67	1.67	1.43	1.19	1.67

2.4. Performance Evaluation Results

After weighing the importance of selected parameters and rating each SDP, results suggests that Spiral model is the most suitable (with the score of 86.76). See figure 1for scores.

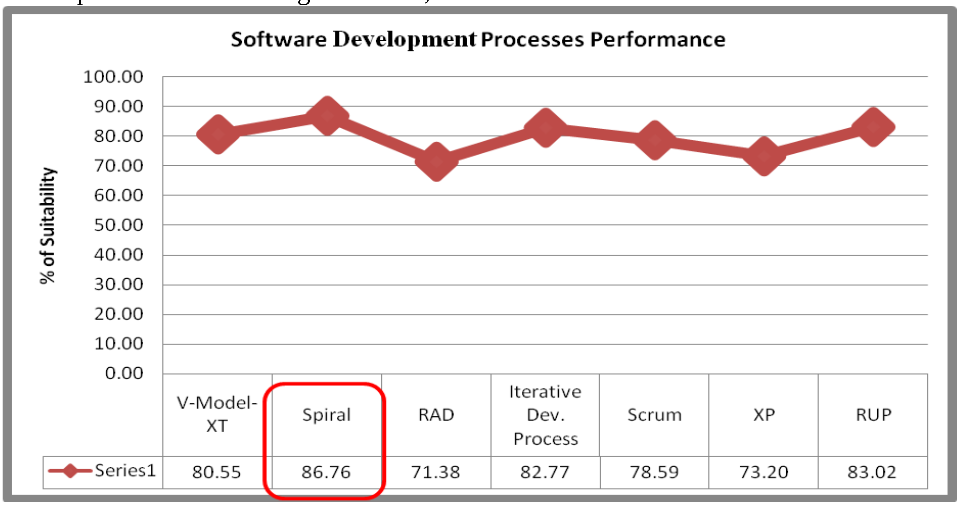

Figure 1: Total criteria scores for the seven SDPs.

3. RESULTS ANALYSIS

3.1. Spiral results Analysis

The results make sense given that the automotive industry software engineering has to produce safety critical features. Spiral Model is suitable for critical systems that need to be adaptable as well. As we can see in figure 2, Spiral model performed very well in almost every category except cost, suitability and efficiency. If cost was rated as the highest importance, then, spiral model will not likely be the most suitable. Finally, although evaluation shows spiral was the most suited based on the prioritized parameters, slight change in parameter priority can results in different software development process.

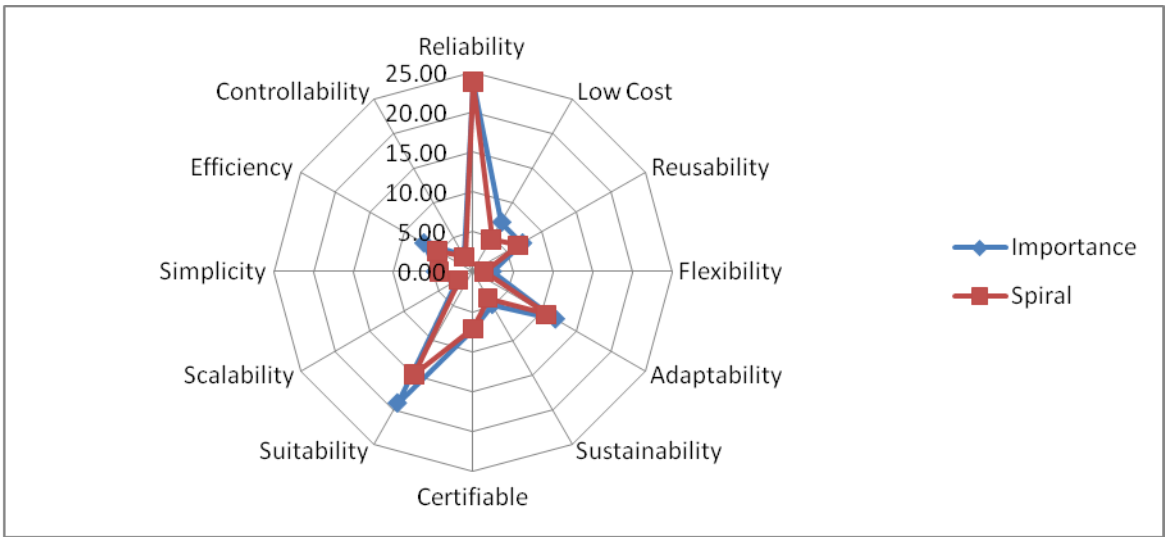

Figure 2: Spiral model performance with respect to the prioritized parameters criteria

3.2. Agile Results Analysis

Another interesting model that we were expecting it to score high is the agile method Scrum. As shown in figure 3, Scrum suffered most from low score on sustainability and certifiable parameters. This is due to agile principles of less documentation and more dependence on individuals. As a result, Scrum development process can threaten sustainability and the ability for an organization to be certified. Having said all that, I think that this method can still be followed on less-critical projects. In addition, documentation can be made ready for certification purpose and train multiple key associates to ensure sustainability.

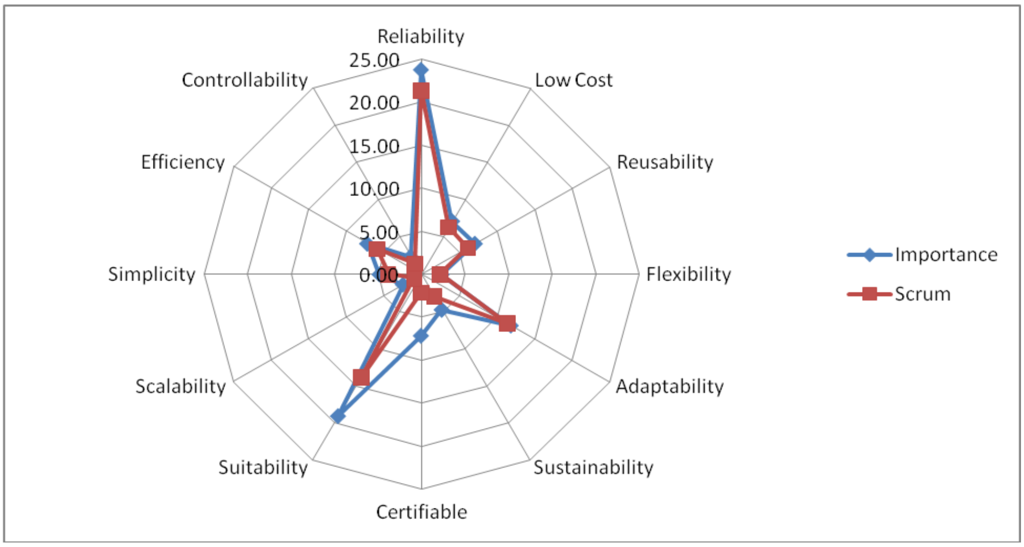

Figure 3: Scrum model performance with respect to the prioritized parameters criteria

3.3. *Key Parameter against all SDPs* (adaptability vs All SDPs)

Adaptability is one of the key parameters for evaluating SDPs for automotive industry. Figure 4 shows SDPs performance against Adaptability parameter. All SDP models, except V-model, are adaptable to new changes. This is because they allow for software to be released iteratively and incrementally. V-model suffered most from adaptability parameter. Agile methods were rated the highest due to their adaptability to new and frequent changes.

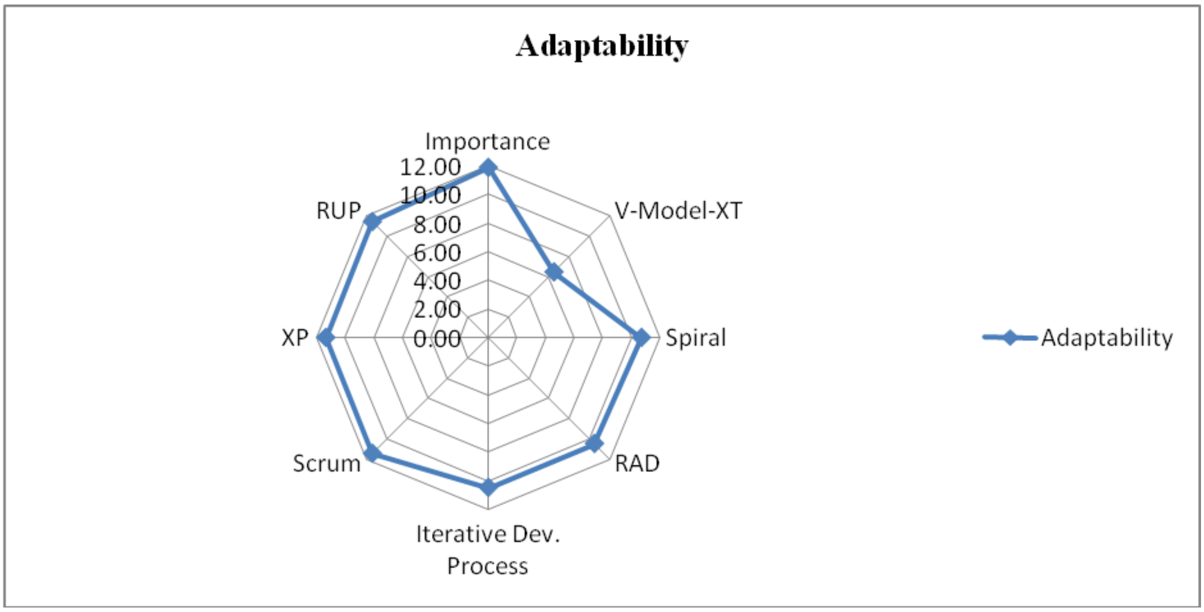

Figure 4: Adaptability score for all the seven SDPs.

4. CONCLUSION

Selecting a suitable software development process (SDP) for the automotive industry software engineering requires careful selection of prioritized parameters for evaluation. In this research many SDPs are evaluated against a prioritized list of parameters. Based on the evaluation results, spiral model scored the highest. High score in all of adaptability, certifiability, and reliability was the key difference for the spiral model over the other selected models. Agile methods scored low on certifiability, whereas, V-model scored low on adaptability. Although spiral model scored highest, it still shows some deficiencies in dealing with cost pressure. So, there is a chance for future research to come up with a suitable SDP that overcomes all challenges including cost reduction. Finally, although evaluation shows spiral was the most suited based on the prioritized parameters, slight change in parameter priority can results in different software development process.

5. FUTURE RESEARCH

5.1. Study the sensitivity of the performance evaluation to improve accuracy of the evaluation.

5.1.1. This allows us to know which parameters are most sensitive to the results. Using this data, we can add more focus on those parameters when weighing the importance and rating the SDPs. This will improve accuracy of the evaluation.

5.2. Study how dividing the automotive industry into several unique domains can help select more suitable processes.

5.2.1. Dividing the automotive industry into several unique domains can help select more suitable processes. So, it's worth the study as this approach can help improve overall quality, delivery and cost for each domain in a different way.

5.3. Hybrid software development processes

5.3.1. With different processes scoring high in different areas, it's worth to study the feasibility and performance of combining multiple development processes.

5.4. In addition to suitable Software Development Process, we can study other factors to improve automotive industry software engineering challenges

5.4.1. System architecture

5.4.2. Software Architecture

5.4.3. Requirements modeling to improve complexity

5.5. Software engineering cost reduction

5.5.1. Cost pressure is another challenge facing software engineering in the automotive industry. So, reducing cost is another key element of business sustainability.

REFERENCES

[1] Broy, Manfred "Challenges in Automotive Software Engineering";

[2] Kumar, G.; Bhatia, P.K. "Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies", 2014, Page(s): 189-196

[3] Terry Bahill, University of Arizona "The Decision Analysis and Resolution (DAR) Process" terry@sie.arizona.edu, 2005. http://www.sie.arizona.edu/sysengr/slides/

[4] Hamed, A.M.M; Abushama, H.; "Popular agile approaches in software development Review and analysis" 2013, Page(s): 160- 166

[5] Taromirad, M.; Ramsin, R. "An Appraisal of Existing Evaluation Frameworks for Agile Methodologies": 2008, Page(s): 418- 427

[6] Taylor, M.A.; Urban, J.E "A method for evaluating software engineering environments": 1994, Page(s): 445 - 450

[7] Salo, O.; Abrahamsson, P. "Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum" 2008, Page(s): 58 - 64

[8] Chowdhury, A.F.; Huda, M.N. "Comparison between Adaptive Software Development and Feature Driven Development" 2011, Page(s): 363- 367

[9] Victor R. Basil, University of Maryland "Quantitive evaluation of software Methodology". https://www.cs.umd.edu/~basili/publications/technical/T47.pdf. 1985.

[10] Layman, L.; Williams, L.; Cunningham, L. "Exploring extreme programming in context: an industrial case study" 2004, Page(s): 32- 41

[11] El-Haik and Adnan Shaout, Software design for six-sigma – A roadmap for excellence. John Wiley, ISBN 978-0-470-40546-8, 2010.

[12] Adnan Shaout and Cassandra Dusute, (2013)," ResPCT – A new Software Engineering Method", International Journal of Application or Innovation in Engineering & Management (IJAIEM) 12/2013; Volume 2(Issue 12): Page 436 – 442, Impact Factor: 2.379.

[13] Wikipedia, https://www.wikipedia.org/

[14] Gamal Waza and Adnan Shaout, "Solutions to Automotive Software Engineering Challenges" technical report number software-engineering-methods-12/14for Dr. Adnan Shaout at the University of Michigan – Dearborn, Fall/2014.