A NEW PROCESS MODEL FOR EMBEDDED SYSTEMS CONTROL FOR TELECOM INDUSTRY

Sanjai Gupta[1],
[#]Department of Information & Technology,
Nizwa College of Technology, Nizwa, Sultanate of Oman
[1]guptasanjay3@gmail.com

Mohammed Hussain[2]
Department of Computer Science & engineering,
M G Institute of Management & Technology,
Lucknow, UP, India
[2]mohd.husain90@gmail.com

***Abstract*— This research deals with important issue for the embedded system in telecom industry. The rapid increase of a software and software based functionality brings various challenges for the telecom industry. As we all know for any given software product the most important thing is its cost, reliability, schedule and quality, these all can be achieve by following good software process models. And hence most organization and businesses put more emphasis on software processes by asking their software engineers to follow it. This paper is to present a modified software process model by mapping V-Model with Personal software process (PSP), Team software process (TSP) and Six Sigma.**

***Keywords*— Software Process Models, Embedded Systems Control, Telecom Product, Quality, Reliability**

I. INTRODUCTION

Today software in telecom products is a dominant factor for the Telecom industry and these software solutions involve large investments and critical data of the organization concerned. Software as well as hardware became enabling technologies in telecom product. They enable new features and functionalities. In a big organization for a given product usually there is lots of different people work within a group for which an organized effort is required to avoid replication and get a quality end product. Many teams work with one goal to deliver final telecom product. Efforts are required to coordinate software, hardware and systems among these teams. In order to achieve such goal a structured process is required that is not to complicated yet very efficient.

II. QUALITY, RELIABILITY, COST & SOFTWARE PROCESS

A. Quality

A quality can be defined as a fuzzy linguistic variable since quality can be very subjective. It can be defined with respect to an attribute such as cost or reliability. For example, a product should be reliable or a product should be both reliable and useable. Similarly product should be affordable, efficient, and effective. *[1]*

B. Reliability

The IEEE defines reliability as "The ability of a system or component to perform its required functions under stated conditions for a specified period of time. *[2]*

C. Reliability as a Quality Product

ISO 9126 defines six quality characteristics, one of which is reliability. IEEE STD. 982.2-1988 states "A software reliability management program requires the establishment of a balanced set of user quality objectives that will assist in achieving the user quality objectives". Since reliability is an attribute of quality, it can be concluded that software reliability depends on high quality software. *[3]*

D. Cost as a Quality Attribute

Quality is always deemed to have a direct relationship to cost – the higher the quality standards, the higher the cost. Quality may in fact have an inverse relationship with cost in that deciding to meet high quality standards at the beginning of the project/operation may ultimately reduce maintenance and troubleshooting costs in the long term. *[3]*

E. Software Process

There are many software processes. Below is the list of process method that are either in use or were used in past, for various types of projects in different industry.

i. PSP and TSP
ii. Waterfall Model
iii. V-Model & V-Model XT

iv. *Spiral Model*
v. *Chaos Model*
vi. *Top Down and Bottom Up*
vii. *Six Sigma*
viii. *Model Driven Engineering*
ix. *Iterative Development Process*
x. *Agile Software Process*
xi. *extreme Process (XP)*
xii. *LEAN method (Agile)*
xiii. *Wheel and Spoke Model*
xiv. *Constructionist Design Methodology*

III. V-MODEL, PSP, TSP AND SIX SIGMA

In the later stage of the paper PSP and TSP are mapped to V-Model. And hence some basic aspects of these processes are below:

A. *V-Model*

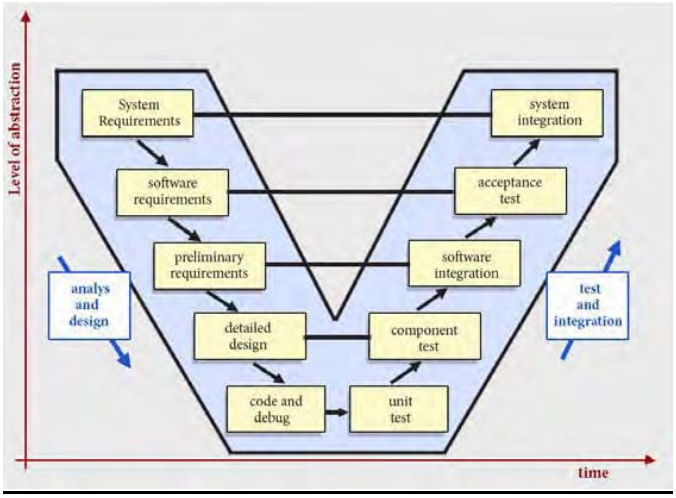
As we know The V-Model is designed for developing system that includes both software and hardware that is why it is mostly used for embedded software in telecom industry.
V-model is very easy to understand and use. It is practically oriented.it answered three basic questions.

- *What has to be done?*
- *How is it done?*
- *What is used to do it*

Based on these questions general structure of V-model mention below which encompasses three levels:

1. The life cycle process model *(procedure):* In this all products can be seen as living products. They are refined, updated and detailed thought out the activities. They can be modified and worked further on only if the version number is updated.
2. The Allocation of methods *(method*): Methods can be dynamically added and removed in the V-model. This model is very flexible in this sense.
3. The Functional Tool Requirements *(Tool requirements):* Any time tools can be reviewed and new emerging tools can easily be added to the model.

At all the levels the standard is structured according the areas of functionality. These areas of functionality are called sub models:
Project Management (PM): This sub models plans, controls and monitor the Project. It also passes information to other sub models.
System Developments (SD): This sub model develops the system or software.
Quality Assurance (QA): This sub model specifies the quality requirements and informs the other sub model of it. This sub model is about assessing product and processes.
Configuration Management (CM)-This sub model administrates the generated products. *[4]*
For each new started project the V-model is tailored into project specific V-Model. When a project finished the final project report is put in the organization experience database. It is the recording of project history in the CM sub model and the general data dictionary of the organization. The general data dictionary contains terms and definitions that are used across whole organization. It is specific for the business domain. Both the project and organization benefit from it.
The V-Model named V-model has to do with the activity and verification flow of the SD sub model. It can be modelled as V in *figure-1*. *[4]*



V-Model Figure-1

B. *PSP-Personal Software Process*

The Personal Software Process (PSP) is an SEI technology developed by Watts Humphrey that brings discipline to the practices of individual software engineers, dramatically improving product quality, increasing cost and schedule predictability, and reducing development cycle time for software. The PSP process consists of a set of methods, forms, and scripts that show software engineers how to plan, measure, and manage their work. PSP is a lightweight CMM level 5 processes designed for cost effective individual use. It applies to most structured software development tasks including requirements definition, architecture design, module development, and documentation production. It is capable of efficiently producing very high quality software products. There is no cost overhead involved in achieving these high software quality levels. In fact PSP projects are generally faster and cheaper than more conventional approaches to software development.
The PSP project was aimed at demonstrating that a CMM level 5 processes could be used by an individual to develop high quality software without excessive process overhead.
The structure of the PSP process is shown in *figure-2* .Starting with requirement statement, the first step in the PSP process is planning. There is planning scripts that guide this work and a plan summary for recording the planning data. While the engineers are following the script to do the work, they record their time and defect data on the time and defect logs. At the end of the job, during post-mortem phase (PM), they summarize the time and defect logs, measure the program size, and enter these data in the plan summary form. [*5]— [7]*

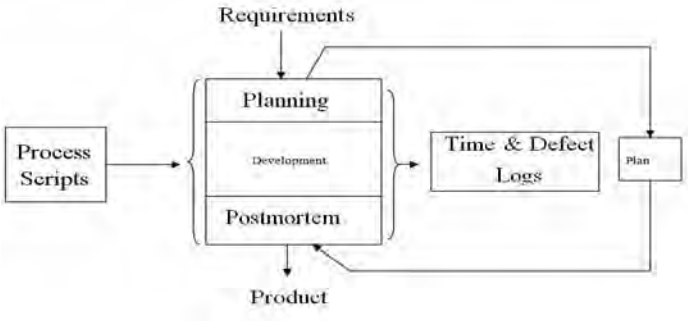


Figure-2: PSP Process Flow

C. *TSP-Team Software Process*

The Team Software Process (TSP) is a complementary SEI technology that enables teams to develop software-intensive products more effectively. TSP was developed to provide a framework for applying PSP in a team setting to develop high quality software. TSP shows a team of engineers how to produce quality products for planned costs and on aggressive schedules. TSP adds a project management layer to the PSP. It helps engineers to produce quality products for planned costs and on aggressive schedules. It addresses the CMM level 2 & 3 management processes using high performance inter-disciplinary work teams. Engineers manage their own work and take ownership of their plans and processes. TSP helps the engineers to build a gelled, self-directed team and to perform as effective team members. It shows management how to guide and support these teams and how to maintain an environment that fosters high team performance.

PSP augmented by TSP can support the development of large-scale software systems. It can be used to accelerate an organization from CMM level 2 to level 5. It provides an excellent foundation for application of six sigma statistical tools.
TSP has been used with pure software teams and with mixed teams of hardware, software, systems, and test professionals. It has been shown to sharply reduce total development cost. *[8]*

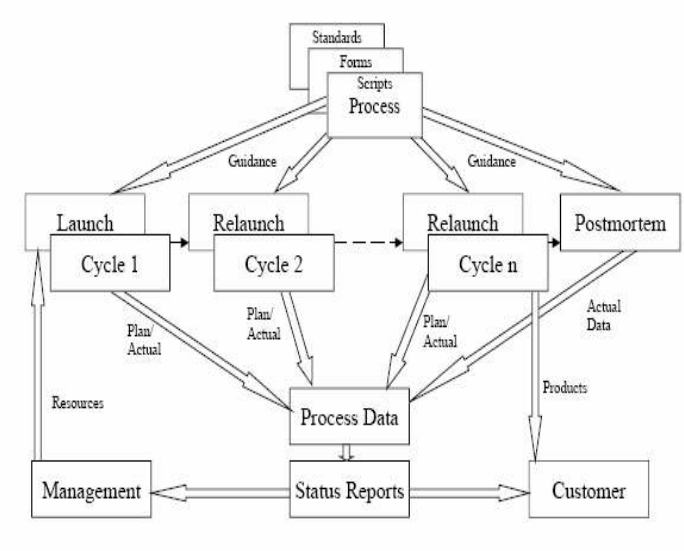


Figure-3: TSP-Process Flow

D. *Six Sigma*

Six Sigma is a philosophy of doing business encompassing the methodologies of continuous improvement, statistical process control, and defect prevention. Six Sigma uses a metrics driven approach to continuous improvement that starts with understanding business objectives. Six Sigma demonstrates a quantitative connection between process improvement and business goals. Six Sigma techniques can produce a software product of predictable quality at a predictable cost that meets the business and customer objectives. Six Sigma can be used by software organizations at any level of CMM maturity.

1) *Six Sigma – DMAIC*: Originally, the model was mostly concerned with problem solving to enhance processes by reducing defects and variation that would cause customer dissatisfaction. The Six Sigma Continuous Improvement cycle.

- ✓ Define the process
- ✓ Measure the process
- ✓ Analyse the process to identify causal variables.
- ✓ Improve the process:
  - Modify the process
  - Measure the modified process
  - Verify the improvement
  - Define control mechanism
  - Control the process to new performance levels
  - Monitor performance metrics & take designated action when
  - required
  - Perform continuous verification of the stability & capability of the process.

2) *Using Six Sigma Tool Kit:* Six Sigma provides a set of tools making the process clear and structured and therefore easier to proceed through in order to save both time and effort and get sooner to the goal. Some of those tools are mention in *table-1*. *[10]- [11]*

Define & Measure	• Project Charter • Risk Analysis (RA) • Statistical Process Control (SPC) • Fault Tree Analysis (FTA) • Failure Mode Effects Analysis (FMEA) • Voice of the customer • Measurement System Evaluation (MSE) • Process Flow Chart • Process Sigma Calculation
Analyse	• Failure Mode Effects Analysis (FMEA) • Histogram • Pareto Chart • Root Cause Corrective Action (RCCA) • Process Map Review and Analysis • Statistical Process Control (SPC) • Regression Analysis • Analysis of Variance (ANOVA) • Design Of Experiments (DOE)
Improve & Control	• Statistical Process Control (SPC) • Failure Mode Effects Analysis (FMEA) • Design of Experiments • Simulation Software • Control Chart • Cost Saving Calculation

Table-1: Six Sigma Tools

IV. INTEGRATING PSP, TSP AND SIX SIGMA

PSP & TSP provide an ideal foundation for the application of Six Sigma techniques to software development. PSP and TSP provide a stable process and high quality metrics for the application of Six Sigma tools. Six Sigma complements PSP and TSP by providing a statistical analysis toolkit including

- Correlation Analysis
- Analysis of Variance (ANOVA)
- Failure Modes Effect Analysis (FMEA)
- Statistical Process Control (control charts)
- Control Plans

Six Sigma toolkits is used to

- perform effective post mortem data analysis,
- measure the effect of process changes,
- control process performance to planned values

DMAIC model “wraps around” the PSP process providing a structure mechanism for continuous improvement at the personal and team levels.
[9]- [10]

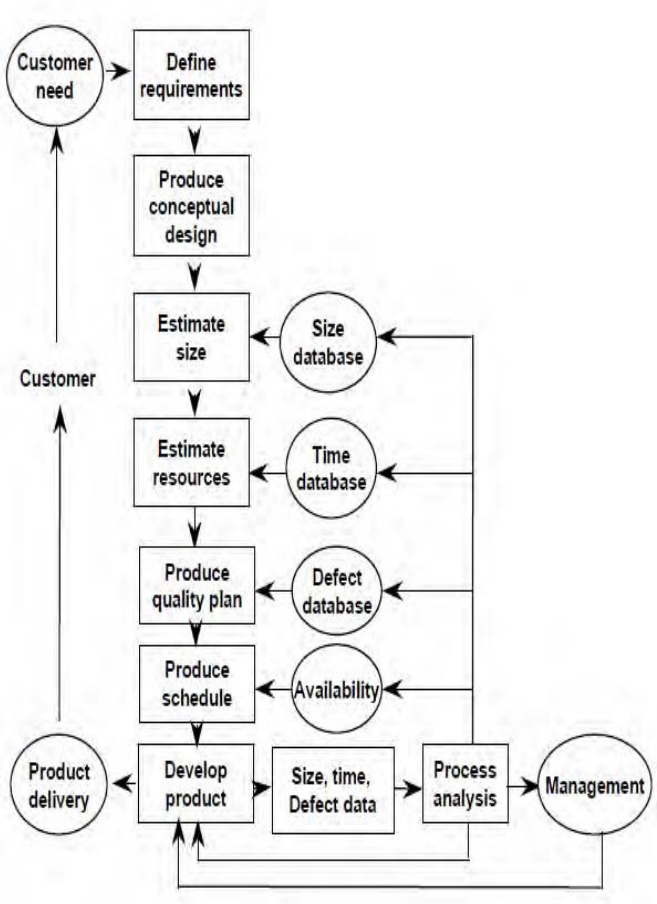


Figure-4: A Closed Loop Process

V. MODIFIED V- MODEL BY USING PSP, TSP AND SIX SIGMA FOR EMBEDDED SYSTEMS CONTROL

A. *What Is Needed?*

The need of a software process model which is efficient, easy to implement, less complex with minimum resource loading that could be used for small, simple, medium, large, and or very complex embedded hardware, software and system control project.

B. *Why V-Model need to mapped with PSP, TSP & Six Sigma*

- V-model is a life cycle project process model. It is only used once during project does not address the whole organization.
- The V-Model argues that the sub models cover all the activity areas, but they do it on a too abstract level. It is hard to find out if peer reviews and inspection are done in The V-model.
- There is no organizational institutionalization of the processes in the V-Model. The goal of a V-model is to successfully reach the goal of a project.
- As the V-model is a project process model not an organization process model, There are no long-terms plans on continuous process improvements.
- I think that any organization use embedded software, that uses The V-model, would need a complementary process model on the organizational level.

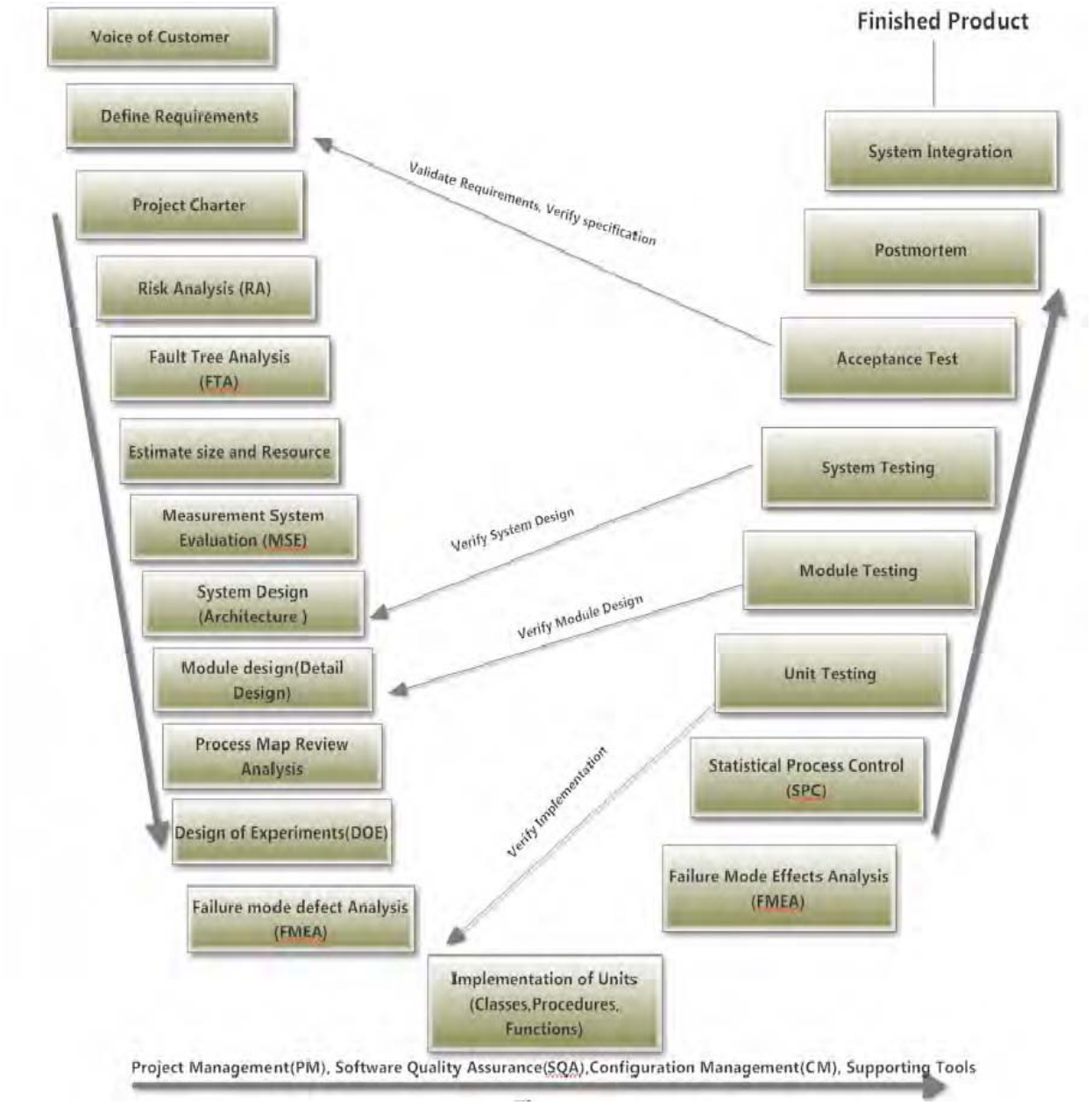


Figure-5: Modified V-Model using PSP, TSP & Six Sigma

C. *PSP, TSP & Six Sigma Mapped with V-Model*

In above *figure- 5* we have modified the V-Model by using PSP, TSP and Six Sigma. The unique difference to the processes discussed before is that use of analysis tools like Risk Analysis (RA), Project Charter, Fault Tree Analysis (FTA), Measurement System Evaluation (MSE), Failure Mode Effects Analysis (FMEA), Design of Experiments (DOE), and Statistical Process Control (SPC) through the various stages of the software life cycle to help determine quantitative and qualitative aspects of Functional, Availability, Reliability, Redundancy, Security, Safety, and Quality. By applying best trade-off approach through an analysis for a given situation for a given technology with good understanding of time and resource required. In the long run it will reduce the cost and over improve the System Quality.

VI. CONCLUSION

This research work explores different software process namely Personal Software Process (PSP), Team Software Process (TSP), V-Model and Six Sigma which further mapped together for Embedded System Control in telecom industry. As we know Embedded System Control consists of hardware, software, and the system control, so that Final Quality depends on the combined quality of software, hardware and system controls and

their design. Thus just to focusing on software quality the overall quality of the system cannot be achieved. While using various analysis method it was found that some decisive requirements were not in place from the start hence it caused ripple effect that were observed in the design, coding, Implementation and testing. For these reasons, V-Model using PSP, TSP, and Six Sigma modified to cover all aspects of software, hardware and system control for embedded system control.

REFERENCES

[1] Dr. Adnan Shaout and Dr. B. EL-Haik, Software Design for Six Sigma: A roadmap for excellence, 1st ed. To appear at John Wiley Press, 2008

[2] IEEE Standard 982.2-1987 Guide for the Use of Standard Dictionary of Measure to Produce Reliable Software.

[3] Gillies, A. C., Software Quality, Theory and Management. United Kingdom: Chapman Hall Computing Series.

[4] Christian Bucanac, "The V- Model" IDE, University of KarlskronaRonneyby, 1999...

[5] Ferguson, P., Humphrey, W., Khajenoori, S., Macke, S., and Matvya, A. "Introducing the Personal Software Process: Three Industry Case Studies," IEEE Computer, 30, 5 (May 1997): 24-31.

[6] Humphrey, W. "The Software Quality Index," *Software Quality Professional*, 1, 1 (December 1998): 8-18M.

[7] Watts S. Humphery, "The Personal Software process", Software Engineering Institute, Nov 2000.

[8] Watts S. Humphery, "Introduction to Team Software Process, Addision- Wesley Professional, 1999.

[9] Steve Janiszewski, Ellen George, "Integrating PSP, TSP & Six Sigma" Software Process Improvement University of Maryland Conference Center , 2002. *www.softwaresixsigma.com/PDFs/0211%20IcpsiSix SigmaPSP.pdf*

[10] Dr. Stephen Thomas Janiszewski, PSP, TSP and Six Sigma at Honeywell International, March 2000

[11] Steve Janiszewski, "Six Sigma & Software Process Improvement". 2003 http://www.softwaresixsigma.com/