

Systemic Comparison of the Application of EVM in Traditional and Agile Software Project¹

By Sam Ghosh

Abstract

This paper reviews the differences between the traditional waterfall software project and agile software project. It identifies the approaches for applying Earned Value Management (EVM) for both waterfall and agile project. It analyzes how an agile software project can overcome the inherent pitfalls of EVM that are also applicable in waterfall project.

1. Introduction

EVM is predominantly used in traditional software projects that use waterfall approach. Literature review shows that the adoption of EVM is not widespread in agile software projects. There is also a misconception that EVM cannot be applied to the agile projects because, the agile projects rely on progressive elaboration. This paper analyzes how EVM can be applied to the agile projects.

2. Key Characteristics of Traditional Waterfall Software Development

The traditional software application is known to follow waterfall software life cycle development. It follows sequential steps of analysis, design, development, testing and rework, implementation. A subsequent step is not started, until the previous step is completed. There is no overlap between these sequences. A thorough analysis of the entire software application is done during the analysis phase. Design is done only after thorough analysis is done. Software coding starts only after all the functions are analyzed and designed. Software testing begins after all the software features are coded.

If a defect is identified in a later stage e.g., during testing or implementation stage, it is costly to fix it as it may require re-work of previous steps of analysis, design and coding.

Requirements are gathered up front. For complex projects spanning months, requirements may change mid-way through the project. Change in scope is costly to

¹ This paper was prepared for course ENCE667 Project Performance Management, a core course for the Master of Engineering in Project Management degree in the A. James Clark School of Engineering, University of Maryland (UM), College Park, Maryland, USA. Course instructor was Dr. Gregory Baecher, Professor and Director of Research of the Project Management Center of Excellence at UM. This paper was originally published in the February 2012 edition of *PM World Today* and is republished here with permission of the author.

incorporate in waterfall project as it will require re-work of many of the previous stages of development.

The milestones and Work Breakdown Structure (WBS) for the entire software application is decided early in the project during the planning process group.

The Figure 1 shows WBS of an Inventory management System. All the work packages are identified during the planning phase.

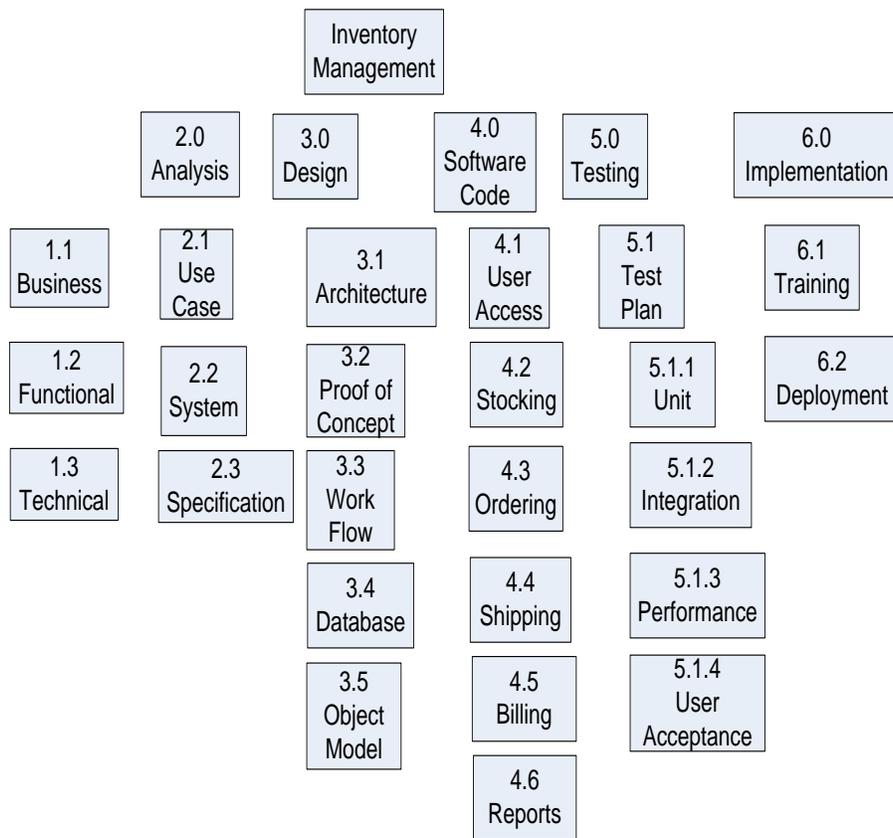


Figure 1. WBS of an Inventory Management Software Application using Waterfall Approach

3. Key Characteristics of Agile Software Development

The software application is broken down into a list of features. These features are completed in different iterations depending on their priorities. WBS is created for the features that are in the current iteration. Each iteration lasts for a few weeks e.g. 2-4 weeks. The customer receives a set of functionalities at the end of each iteration, instead of waiting for months and years till receiving any working software.

Analysis, design, development, testing and implementation are done for each functionality that are in the current iteration.

Each feature is broken down into user stories. Story point is assigned to each user story. Higher the level of effort, higher is the story point. Each story is independent of each other. Customer may decide which story needs to be implemented in a given iteration. It is possible to mix story from different functionalities in a single iteration. The total number of story points implemented in the iterations typically remains the same.

Agile methodology acknowledges that the requirements and the scope of the project will change. In an agile project, change requests can be incorporated in the future releases in a matter of months, or weeks. Agile project can accommodate these changes because of the short release cycle.

The inventory management system shown in Figure 1 has six primary features, which are namely:

1. User Access
2. Stocking
3. Ordering
4. Shipping
5. Billing
6. Reports

Figure 1 is restructured in Figure 2 to show the agile software development. Figure 2 breaks down two of the features from Figure 1 into user stories used in agile projects. Each user story is given a story point.

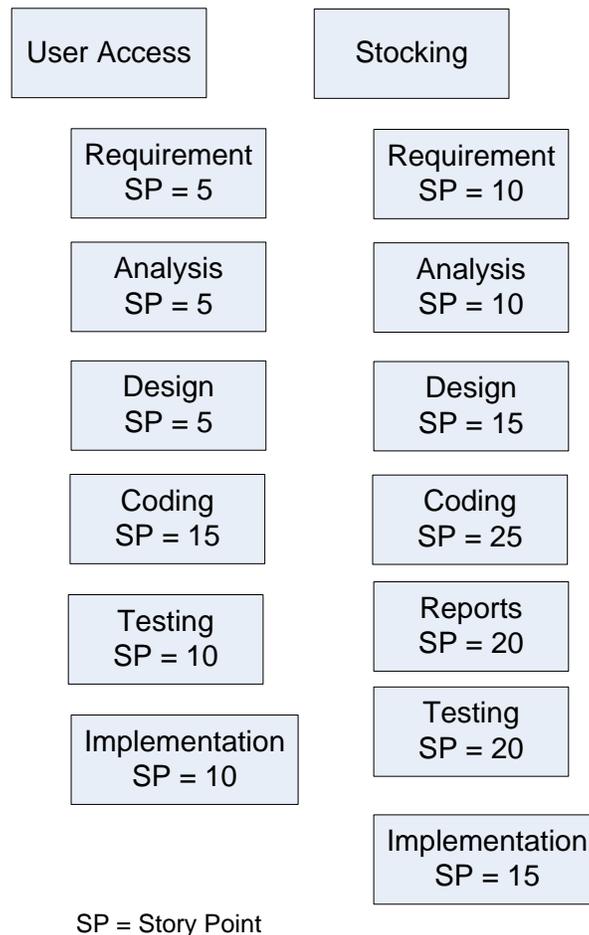


Figure 2. Features of an Inventory Management Software Application broken into User Story using Agile Approach

4. Key Differences between Waterfall and Agile Methodology:

In agile project, the application is broken into features. Analysis, design, coding, testing and implementation are done feature by feature. On the contrary, in the waterfall approach, upfront analysis of the entire application is done before design is started and the stages are sequential. Agile project uses progressive elaboration while the waterfall project sets the base plan early in the project. Customer collaborates in agile project and requests scope change. Waterfall project sets the scope up front and it is costly to make scope changes.

5. Brief Review of EVM

5.1 History of EVM

EVM started in the United States (US) in the 1960s for US federal projects. Today, EVM is widely used on the US federal projects. It is mandatory for the federal projects that have a budget of 20 million US dollars.

EVM measures the cost and schedule of a project. EVM requires a clear definition of the project scope. In order to calculate EVM accurately, it is important that the WBS is accurate.

5.2 Key Benefits of EVM

EVM is used to measure the schedule and budget performance index of a project at a given time. These indices are measured in monetary units. EVM helps forecast the completion date and cost at any given time in the project. It allows the project managers to make decisions based on these metrics. It encourages the use of percent completion of a work package.

5.3 Limitations of EVM

EVM has a few inherent limitations:

1. EVM does not track the scope of the project - EVM does not measure the quality of the deliverables and the technical performances. It only accepts the deliverables upon completion.
2. Schedule Performance Indicator (SPI) is an inaccurate measure during the later part of the project, especially if the project is behind schedule. At the later part of the project, SPI gets close to 1, although the project is over schedule.
3. EVM ignores critical path. As such, it is possible to be ahead of schedule while there is negative float on the tasks on the critical path.
4. EVM costs time and money. Considerable time is spent in tracking these metrics.

6. Application of EVM in a Waterfall Software Project

WBS for the entire software application is created. More accurate the WBS, more accurate will be the EVM. Percent complete for work packages is used for calculating EVM. Percent complete may be subjective. It is not unusual that remaining 10% of the task in a work package takes longer than what it was planned for.

The following metrics are calculated in traditional waterfall project:

- ❖ Planned Value (PV), or Budgeted Cost of Work Scheduled (BCWS): This is the baseline that is planned for the entire application at the planning phase. It is measured in monetary unit.
- ❖ Actual Cost (AC), or Actual Cost of Work Performed (ACWP): Cost that is incurred, measured in monetary units.
- ❖ Earned Value (EV), or Budgeted Cost of Work Performed (BCWP): It is the value of the work performed, measured in monetary units.
- ❖ Schedule Variance (SV) = EV-PV
- ❖ Schedule Performance Index (SPI)= EV/PV
- ❖ Cost Variance (CV) = EV – AC
- ❖ Cost variance Index (CVI) = EV / AC

Above indicators can be used to forecast Estimate At Completion and to project the completion date in a waterfall project. All the limitations of EVM are applicable in waterfall methodology.

7. Application of EVM in an Agile Project

7.1 Features of Software Application are Implemented in Multiple Iterations Based on Priorities

Features of the software application are chosen to be implemented based on their priorities. Features with higher priorities are implemented in prior iterations. Priorities of the features are decided based on one or more factors:

- 1) Dependency: If a feature depends on another feature, assign higher priority to the preceding feature.
- 2) Risk: If a feature has known or unknown risks, or technical challenges then assign higher priority to that feature.
- 3) Value: If a feature provides a higher value to the users, assign higher priority to that feature.
- 4) Customer Request: A customer may request to implement a particular feature in an iteration. Customer collaborates with the software development team in an agile project. After all, customer is the one who is paying for the project. Customer may choose to re-assign priorities on the features to account for any change in scope and requirement.
- 5) Procurement Lead Time: If a feature requires a tool that needs to be ordered or purchased, prioritize it to accommodate for the procurement lead time.

Add the story points of the features that will be implemented in the current iteration. The total point is also known as Velocity. Velocity may differ from one iteration to another, although the duration of iteration is the same. EVM in an agile project can be calculated separately on every iteration.

7.2 Agile Project Expects the Initial Scope to Change

In waterfall methodology, the project scope is defined upfront. On the contrary, agile project expects that the scope of the project will change with time. Agile project is well equipped to incorporate changes in scope through short iterative release cycles. The initial scope of the project may change because of:

- A. Change in business strategy
- B. A feature may become less valuable, as the feature that is already delivered 80% of the functionalities. For example, after implementing the stock reporting system, it was realized that 80% of the reporting needs for the billing reporting system can be extrapolated from the stock reporting system. As such, the priority of implementing billing reporting system was lowered.

As the scope changes in agile project, EVM calculation uses the base plan that was set at the start of the iteration as the baseline.

7.3 An Approach to Calculate EVM in Agile Software Project

EVM concept used in waterfall project is equally applicable in an agile project. However, the interpretations could be different. Following interpretations are made for calculating EVM in agile project:

- AC: Cost incurred to date in an iteration, measured in monetary unit.
- EV: Value delivered to the customer, measured in monetary unit. It can be measured by adding the story points delivered to the customer, or by adding the dollar value of the functions delivered to the customer. Agile project accepts the user stories that are 100% complete. This is in contrast with waterfall project which accepts percent complete for a work package.
- PV: Value planned to be delivered to the customer, measured in monetary unit. PV can be measured by adding the story points that were planned to be delivered, or the dollar value of the features that were planned to be delivered.

7.4 Charts used in Agile Project

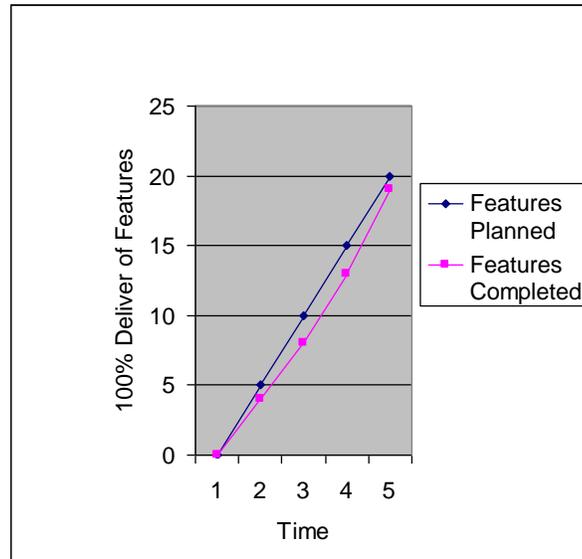


Figure 3. Burn Up Chart using 100% delivery of feature

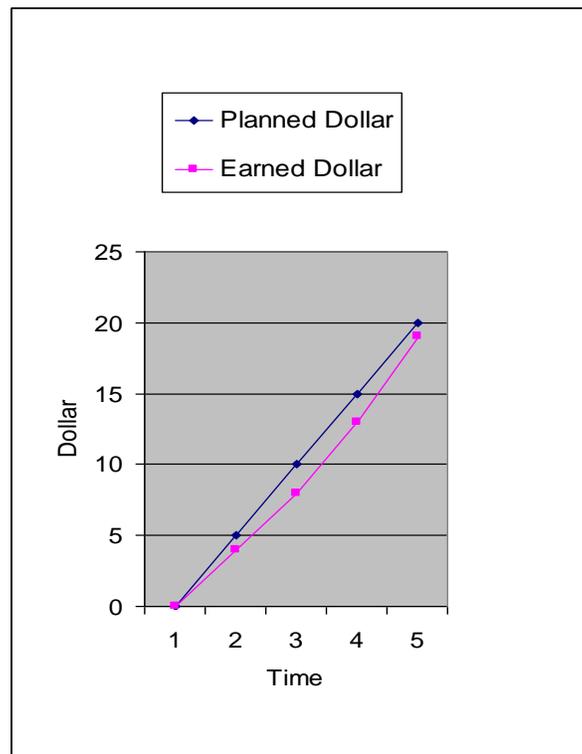


Figure 4. Burn Up Chart using dollar value

Figure 3 shows the Burn Up chart by comparing the features that were planned to be delivered in an iteration with the features that are actually delivered. Figure 4 displays a different view of the Burn Up chart for the same iteration where graphs were drawn by comparing dollar that was planned to be spent versus the dollar that was earned by delivering the features to the customer.

The Planned Feature shown in Figure 3 and the Planned Dollar shown in Figure 4 are synonymous to PV in EVM used in waterfall project. Features Completed in Figure 3 and Earned Dollar in Figure 4 are synonymous to EV. Agile project accepts the features that are 100% complete for drawing Burn Up chart. This is in contrast with percent complete rule of work package used in waterfall project for calculating EVM. The 100% completion rule in agile project is more accurate of an indicator of the project metrics.

The difference between planned features and the completed features in Figure 3 at any point in time in an iteration indicates variance, which is synonymous to SV used in EVM of waterfall project. In the same way, difference between Planned Dollar and Earned Dollar in Figure 4 at any point in time in iteration indicates variance, which is synonymous to CV used in EVM of waterfall project.

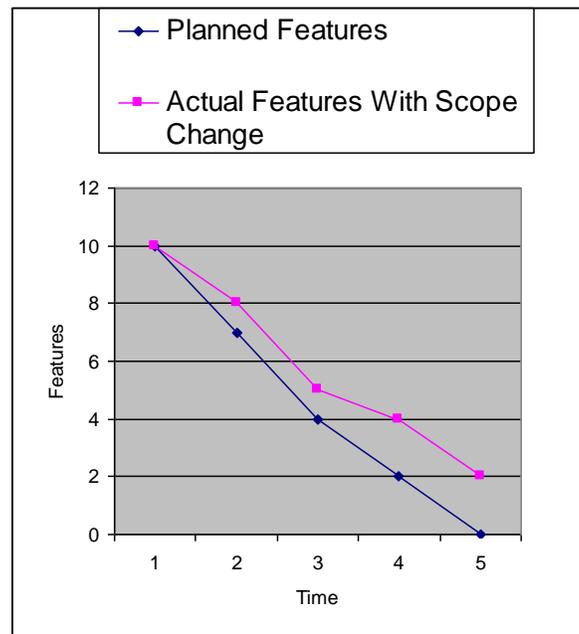


Figure 5: Burn Down Chart

Burn Down Chart in Figure 5 shows the number of features that are yet to be implemented. This chart also shows the change requests made on the features planned in an iteration. The change requests are typically implemented in future iterations so that the features to be delivered in the current iteration are still delivered within the time box of the current iteration.

7.5 Forecasting in Agile Project

Using the interpretations of PV, AC and EV described in section 7.3 along with the Burn Up and Burn Down charts it is possible to forecast Estimate At Completion and Projected Completed data.

8. Agile Project Overcomes the Limitations of EVM that are Applicable to Waterfall Project

The limitations of EVM apply to waterfall project. Agile project overcomes many of the limitations of EVM because of short release cycles, customer collaboration, incorporation of change in scope, elaborative development and incremental delivery of the features in iterations.

8.1 Agile project measures EV based on features delivered to the customer

25% of the overall project could be analysis and design in waterfall project. At the end of the analysis and design phase, several documents are created, 25% of the project is marked as complete and 25% of the project dollar is spent (assuming that the project is on target) but no functionality is yet delivered to the customer. Although Cost Performance Index (CPI) and Schedule Performance index (SPI) may be 1 or positive for the waterfall project at this time, no real benefit is yet delivered to the customer. Agile project overcomes this limitation of waterfall project by measuring EV as the value of the functions already delivered to the customer. At the end of 25% completion in an agile project, multiple iterations will be completed and multiple functions will be already delivered to the customer.

8.2 Agile project prioritizes high risk features which keeps SPI and CPI accurate

In a waterfall project, tasks are executed sequentially. A technical challenge may be encountered late in the project, which may require considerable time and effort that was not planned for. That will impact SPI and CPI in waterfall project. On the contrary, agile project implements the high risk features in the early iteration.

8.3 Agile project aligns the scope of the project for measuring EVM

EVM does not track the scope of the project and it is applicable in waterfall project. EVM in waterfall project is calculated based on the WBS that was created upfront during the early part of the project. On the contrary, agile project expects that the scope will change during the course of the project. Change requests made in an iteration is marked to be implemented in the future iterations. As such, every iteration typically includes a few change requests that were made from previous iterations. EVM measured in an agile project in a given iteration includes the scope changes.

8.4 Agile project incorporates Critical Path in EVM

Waterfall project ignores the tasks on critical path for measuring EVM. It simply calculates percent completed for all the tasks in WBS, irrespective of whether the tasks on critical path are completed or not. As such, in waterfall project it is possible to have positive SPI and CPI indicator while the critical path has negative float. In an agile project, the features with higher priority are included in an iteration. In other words, the features in an iteration are on critical path. Work is done only on those features in the iteration. As such, calculation of EVM in an agile project is done only on the features in that iteration and those features are also on the critical path for that iteration.

8.5 Agile project incorporates quality in EVM

EVM does not measure quality. In agile project, customer may request changes to the functions delivered. These changes will be reworked in a few iterations, in a matter of weeks. As such, quality can be incorporated quickly in the delivered functionalities. EVM in an agile project includes the rework, thus incorporating quality improvement earlier than in waterfall project. Note that the concept of “Technical Performance Measurement” (Ferraro, 2001) can be used to augment EVM measures in both waterfall and agile project. However, the agile project because of short release cycles is more adept to incorporate re-work and quality improvements.

8.6 EVM is calculated for short iterations, which costs less time and money

A waterfall project may last for months, or years. The total number of work packages may count in thousands and budgeted cost could be in millions. It is a daunting task to track it. On the contrary, EVM in an agile project is measured for each iteration, which lasts for only a few weeks with a few features that are implemented. As such, it is easier to track EVM in agile project.

8.7 Rolling Wave Planning of Agile Project

In an agile project, planning is done using progressive elaboration, instead of performing upfront planning for the entire project in the planning phase. Agile project lists the major features in the project upfront. But the exploration of each feature is done based on its priority. As the project progresses, more aspects of the project are known. Cost, schedule and risk estimates become more accurate. Agile project uses this concept of rolling wave planning. Agile project expects the scope to change. As such, it does not have a baseline. This sounds contradictory for using EVM in agile project, as EVM requires a baseline for measuring PV. This is a misconception. This is because, although the agile project uses rolling wave planning, it sets a baseline for the iteration. Iterations in an agile project lasts for a few weeks and PV is of the features to be delivered in that iteration are decided before the iteration starts and that does not change throughout the iteration. The PV for an iteration could be measured by the story points delivered, or dollar value of the story points to be delivered. Instead of deciding

PV for the entire project, agile project decides PV for an iteration and that PV does not change. This fixed PV for the iteration can be used in EVM for that iteration in agile project.

This concept of calculating EVM based on functions delivered can be extended in calculating EVM in research and development projects. In the research projects, there are many unknowns, scope changes, WBS is not accurate. Progressive elaboration can be used in research projects, scope and deliverables can be time boxed in iterations. EVM can be calculated in each iteration in research projects.

8.8 Burn Up Chart of Agile project gives accurate measure of SV in the later part of the project

SV is not an accurate indicator of project metrics in the later part of the project, especially if the project is behind schedule. Waterfall project is prone to this pitfall. As such, waterfall project uses Earned Schedule (ES) in the later part of the project, instead of using SV of EVM. ES measures SV in time, instead of dollar.

Agile project using the Burn Up chart measures SV in terms of features delivered (Figure 3), or dollar value delivered (Figure 4). SV is not measured in terms of time. As such, this measure of SV using features or dollar is true indicator of the project progress.

9. Conclusion

This paper indicates that the concept of EVM is applicable in an agile software project. In many instances the agile project overcomes the inherent limitations of EVM that are applicable in waterfall project. Adoption of EVM will become more popular in agile community with training, education and additional research.

References

- [1] Sliger, M., "Agile project management and the PMBOK® guide", Agile project management PMI Global Congress 2008, North America.
- [2] Sliger, M., Broderick, S., "[Selling agile : how to get buy-in from your team, customers, and managers](#)", PMI Global Congress 2009-North America.
- [3] Project management Institute, "A Guide To The Project management Body Of Knowledge", 4th edition
- [4] Mitchell, S.M.; Seaman, C.B., "A comparison of software cost, duration, and quality for waterfall vs. iterative and incremental development: A systematic review", Empirical Software Engineering and Measurement, 2009, 3rd International Symposium, Page(s): pages(s): 511 – 515
- [5] Sureshchandra, K.; Shrinivasavadhani, J., "Moving from Waterfall to Agile "Agile Conference, [2008](#), Page(s): 97 – 101, Digital Object Identifier: [10.1109/Agile.2008.49](#)
- [6] Sulaiman, T.; Barton, B.; Blackburn, T., "AgileEVM - earned value management in Scrum Projects" Agile Conference, 2006, page(s): 10 – 16
- [7] Alleman, G.B.; Henderson, M.; Seggelke, R., "Making agile development work in a government contracting environment-measuring velocity with earned value", Agile Development Conference, 2003, page(s): 114 – 119
- [8] Cabri, A., Griffiths, M., "Earned value and agile reporting", Agile Conference, 2006, page(s): page(s) 6 - 22
- [9] Bower, D, "PHASE EARNED VALUE ANALYSIS: A PROPOSAL FOR SIMPLIFYING YET ENHANCING EVM" Project Management Institute, 2006
- [10] Alleman, G., Henderson. M., "Making Agile Development Work in a Government Contracting Environment - Measuring velocity with Earned Value" Agile Development, June 2003
- [11] Ferraro, M., "Technical Performance management", Integrated Program Management Conference, November 14-17, 2001
- [12] Kulick, K., "Technical Performance Measurement - The Basics and Beyond", 9th Annual Cost Schedule Performance Management Conference, October 22, 1997

Websites

- [1] <http://agilemanifesto.org/>
- [2] <http://www.scrumalliance.org/articles/18>
- [3] <http://www.expertprogrammanagement.com/2010/03/wbs-work-breakdown-structures-everything-you-need-to-know/>
- [4] http://www.stickyminds.com/pop_print.asp?ObjectId=10631&ObjectType=COL
- [5] DAU Gold Chart, November 20210
<https://acc.dau.mil/CommunityBrowser.aspx?id=19577>

About the Author



Sam Ghosh

Maryland, USA



Soumajit (Sam) Ghosh has over twenty years of experience in industry, and academia in Information Technology, and Project Management. He has worked as Project Manager, and Enterprise Architect for start-up ventures, government agencies and Fortune 500 companies in mission critical projects. He is certified Project Management Professional (PMP), PMI, Sun Certified Enterprise Java Architect, Sun Certified Web Component Developer, Sun Certified Java Programmer and Competent Toastmaster, Toastmasters International. He holds MS in Computer Science, and MS in Industrial Engineering from Texas Tech University. He is a PhD Candidate at A. James Clark School of Engineering, The University of Maryland, College Park, specializing in Project Management. He can be reached at: sgho@aol.com.