How I Learned to Stop Worrying and Love Benchmarking Functional Verification!

Mike Bartley
Test and Verification Solutions
SETsquared Business Acceleration Centre
University Gate East, Park Row
Bristol BS1 5UB, ENGLAND
mike@testandverification.com

Mike Benjamin
Associate at Test and Verification Solutions
SETsquared Business Acceleration Centre
University Gate East, Park Row
Bristol BS1 5UB, ENGLAND
benjamin@testandverification.com

Abstract

This paper describes the 'Functional Verification Capability Maturity Model' (FV-CMM), a benchmarking process to help users measure the maturity of their verification processes and provide a framework for planning improvements.

The most unique feature of the FV-CMM is a well defined methodology linking required capabilities to actual project practices and then quickly identifying the major issues. This is described in the main body of the paper which also describes how the reviewers can then measure process maturity as one of five levels, each corresponding to a clear step in maturity. This allows benchmarking to classify all capabilities into distinct and meaningful categories even without having quantifiable metrics.

Finally the paper describes how benchmarking results may be validated and then used for planning verification process improvements.

Keywords

Benchmarking, Functional Verification, FV-CMMI

Introduction

We are all aware of the need to measure and improve our functional verification processes but getting to grips with benchmarking is actually incredibly difficulty. This paper describes the 'Functional verification Capability Maturity Model' (FV-CMM), a benchmarking process developed by TVS that helps the user to measure the maturity of their verification processes and provides a framework for planning improvements.

The first thing to consider when starting any activity is to clearly define its purpose. Functional verification today faces ever growing challenges. Some are technical resulting not only from the increasing complexity and decreasing timescales of projects but also the need to embrace advances in functional verification technology to remain competitive. Others result from business or organisational changes such as the acquisition or loss of key teams, a move to multi-site working, or a change in the target market such as a decision to move into automotive products. In all these cases we needed to understand how our customers would benefit from applying benchmarking. We identified a number of key aspects:

1. It is essential not only to meet today's challenges but anticipate the future. We sometime see companies that are in crisis because their management has been effectively ambushed by this constant march of technology. Companies need a process that can give them a clear warning before things go wrong!

2. Functional verification requires a vast amount of resources of all kinds: people, machines and EDA licenses. Even more importantly it has a major impact on project timescales. Yet often engineers and management in companies have very different perceptions of current capabilities and fail to identify or address key areas of weakness.

3. A process of continuous improvement needs a shared 'language' and framework that can be used to identify issues, then define, prioritise and monitor tasks. This is a key requirement for companies to ensure they will continue to be able to meet future verification challenges.

Over the years there have been numerous attempts to develop benchmarking methodologies. One approach is to measure progress against known metrics or roadmaps such as the International Technology Roadmap for Semiconductors. This can be very useful for looking at a specific aspect (Meeth, 2010) but the focus on specific aspects with defined targets tends to result in losing the 'big picture'. Alternatively one of the most widely used methodologies is the Capability Maturity Model (CMMI) (Carnegie Mellon University Software Engineering Institute, 2010). Whilst aimed at software engineering it provides a framework that is widely applicable to most business activities. However, whilst we draw considerable inspiration from CMMI, by trying to provide a general purpose framework it, out of necessity, has a number of serious limitations when trying to use it to benchmark a highly specific activity such as functional verification:

1. The CMMI is relatively abstract and does not address domain specific 'capabilities', yet these are at the heart of effective functional verification[1]
2. Deploying CMMI is actually quite an involved process that takes considerable time and expertise. Even reading the specification is quite a lengthy business. Our experience suggested that this was a major barrier to adoption.
3. Function actually follows form. The capabilities of teams are largely shaped by their organisation and practices. Imposing a rigid benchmarking process can over time distort an organisation and prevent necessary change.

Much the same observations have been made independently by other industry experts (Foster & Warner, 6/2009). For the above reasons we aimed to develop a more specific, but flexible and light-weight process dedicated to benchmarking functional verification. The FV-CMM is a framework that provides a light weight solution for benchmarking functional verification capability which can provide:

- An integrated view of the organisation from the viewpoint of functional verification
- An objective benchmark for measuring the maturity of functional verification activities
- A framework for process improvement that can help management define goals and priorities

[1] For this reason software testing has developed the domain specific ´Test Maturity Model Integration' (TMMi)

Whilst it has some similarities to the 'Evolving Capabilities Model' Foster and Warner proposed it has a unique approach to decomposing capability in a 'top down' fashion and then evaluating maturity 'bottom up'. The rest of this article describes the three key elements of this benchmarking process: *capability*, *maturity* and the actual benchmarking *process* that TVS adopts. It also shows how the methodology can make the link between a high level view of capability and the specific verification activities being undertaken on actual projects.

Capability

The FV-CMM benchmark has a hierarchical structure that starts by breaking capability down into key process areas such as *'functional verification planning and scenario creation'*. These can be customised for each client as a company developing interface IP will face different challenges to one developing CPUs or doing SoC integration. The process areas may also change over time as companies evolve and technology continues to develop. The only requirement is that each should have a clearly defined purpose and a clear impact on functional verification. We have so far defined 13 possible process areas ranging from '*functional verification planning'* through '*metrics, coverage and closure'* to '*reviews'.* These are not abstract ideas but specific capabilities required for effective functional verification. They generally fall into two groups: **methodology** being the body of principles and practices used to solve tasks and **process** being a series of actions or operation that produce a specific service or product. Two of the process areas are of particular interest as they do not directly refer to functional verification. One is '*specification and design*' which is the bedrock on which functional verification is built whilst the other, '*organisational capability',* addresses the need to learn and adapt.

Each process area in turn consists of a set of specific goals (eg: '*ensure the integrity of the code base'*) and practices (eg: *'all tasks should have an agreed completion date'*) that capture key requirements (the "what"). For example in the case of '*specification and design'* the specific goals and practices for functional verification are:

- *Give the verification team visibility of the architecture and micro-architecture corner cases*
- *Make the design 'verification friendly'*
- *Make the design stable to ensure verification isn't trying to hit a moving target*

These in turn are broken down into example actions and activities that address that issue (the "how"). These are not intended to be exhaustive but do serve to connect the abstract framework to concrete actions. For example design stability includes *'checking whether the project enforces a process of successively freezing the RTL'*. This structure can easily be customised to the specific needs of different application domains, different design styles or different companies. The resulting framework can be captured on a single spreadsheet, as partly illustrated in Figure 1.

5 System level testing
5.1 The purpose of each test bench should be clearly identified
5.1.1. The purpose and the scenarios to be reached by each test bench are clearly identified. The purpose must consider the appropriate level of testing for the various scenarios (e.g. integration with other IP, software debug features, low power features, performance validation via benchmarking)
5.1.2. Regression testing, using appropriate scenarios and checkers, is used to validate bug fixes and ensure errors are never reintroduced.
5.2 Validate key capabilities essential to early deployment
5.2.1. Architectural verification tests will fully cover the architecture but be design neutral. Device verification tests is design specific.
5.2.2. Tests are self checking to run on multiple platforms including simulation, emulation, FPGA and silicon
5.2.3. It is possible to determine that the checking mechanisms employed by the test bench are sufficient. That is, they are able to detect any bug uncovered by the stimulus.
5.3 Demonstrate the ability to execute key software programs
5.3.1. The verification will include executing software such as operating system bring up and running key customer applications (where practical). The software will also be run in the presence of hardware irritators.

Figure 1: Part of an FV-CMM framework

Maturity

When evaluating maturity we consider three aspects:

Ownership: this can vary from tasks, tools and expertise being specific to named individuals to ownership being shared across the project or the entire company wide community. This corresponds to the level at which: adoption has occurred, decisions are made, or support can sensibly be requested. This also reflects the process for continuous improvement that can vary from best practice being owned by individuals who implement improvements in an ad hoc fashion to institutionalised fact based learning.

Visibility: this can vary from undocumented, with no external input, to living documentation with quantitative metrics and full involvement of the stakeholders. It is related to three aspects: the availability of documentation, the use of metrics for measuring progress and quality, and the use of reviews.

Execution: this can vary from ad hoc working where completion is never checked to a repeatable process permitting planning and fact based continuous improvement. Typical characteristics of a repeatable process are documentation and automation.

The maturity of each aspect is defined as being at one of five possible levels. Each of these levels corresponds to a clear step in maturity. These are:

Initial: Processes are typically ad hoc and applied incompletely or on a best effort basis, especially in times of crisis. Goals are often not satisfied. Processes are typically not documented or otherwise made repeatable and best practice remains in the ownership of individuals rather than being captured by the organization. Verification planning is either not performed or is performed and not documented, or plans are incomplete and not maintained once written. Stakeholders are not normally involved in the planning.

Managed: The processes are performed consistently and the goals are satisfied. Processes are owned and aligned at project level. They are automated, or otherwise repeatable, and will serve to locally capture best practice. However there are few specific checks on the capabilities of tools and processes. Initial verification planning is performed and documented but the plans are not maintained. Metrics are used to demonstrate progress (scenario completion, code coverage, bug rate) but not to check that the plan has been implemented. The status of the work is only visible to management at defined points and the predictability of verification completion is weak.

Defined (also known as ´Planned‘): The processes are planned in conjunction with the relevant stakeholders. Implementation is adequately resourced. The verification plan is either maintained over the life of the project or is a living plan. In either case there are checks or coverage metrics allowing the results to be monitored and reviewed. The capability of specific processes and tools is reviewed qualitatively to ensure good alignment with tasks. The predictability of verification completion is strong. Best practice is consistently shared across projects.

Quantitatively Managed: The organisation is using metrics and profiling. Living documentation ensures full visibility at all times and ensures the widest possible involvement of stakeholders in the verification process.

Optimising: The organisation practices fact based learning and continuous improvement at an institutional level using data collected across the organisation and projects. Quantitative metrics are used for both coverage closure and continuous improvement of product, tools, process and organisation.

Process maturity, as summarised in Figure 2, is not a substitute for skilled and dedicated Engineers. However they will be greatly helped by ensuring that process maturity is appropriate to the current needs of the organisation. A start up or skunk works project with a small team of experts may be best served by having an **initial** process maturity. But as organisations grow increasing process maturity will make the work of those individuals more predictable and repeatable, and make it easier for the organisation to learn from best practice. And it is often fast growing companies undergoing organisational change that most urgently need to address their process maturity.

	Initial	**Managed**	**Defined**	**Quantified**	**Optimising**
Ownership	Individual	Project Team	Project Stakeholders or ad hoc groups of projects	Community	Company wide or institutionalised
Visibility	Not documented. No reviews. No metrics.	Documents incomplete or unmaintained. Point reviews. Progress metrics.	Maintained docs. Continuous tracking against quality metrics.	Living docs. Quantified quality metrics.	Data integrated across the organisation.
Execution	Ad hoc	Tasks performed but completion not explicitly checked	Tasks planned and implemented in a systematic fashion. Check completion of planned tasks.	Quantifiable metrics used for coverage closure and release determinism	Quantifiable metrics used to drive continuous improvement.

Figure 2: Process Maturity

Process

The process areas are largely fixed but no ‘one size fits all’. The specific actions and activities are often organisation, time or even project specific. Hence the first step in applying the FV-CMM is to customise the framework to the target organisation and its objectives. This allows the benchmarking to reflect the needs of the organisation. The results of are then captured in a single spreadsheet.

Evaluation against the FV-CMM benchmark proceeds ‘bottom up’ using the example actions and activities to structure evidence gathering. This typically takes the form of in depth interviews with small groups of key project or department staff including verification managers, design managers and project managers as well as key verification experts. The interviewers work in pairs with one acting as the interviewer and the other as the recorder, though they can swap roles. This helps maintain the pace of the discussion and makes it easier to ensure all topics are adequately covered.

To ensure a productive discussion it is important to engage participants by keeping questions ‘open’ and not forcing the discussion to strictly follow the structure used in the benchmarking spreadsheet. The interviewers may instead engage the participants by approaching the issues in different ways, for example by asking about key themes such as ‘the main sources of complexity in the project’ or ‘release determinism’. The answers will then be mapped back into the benchmarking framework.

The results of the interviews can in turn be backed up by reviewing project documents and data but this differs in subtle ways from an audit. Here the intention is to facilitate discovery and draw out the collective knowledge of the team rather than enforce practices.

The observations are recorded and validated by being fed back for comment to the interviewees and other relevant staff. The reviewers then use their expertise and this evidence to 'score' the maturity of each of the three key aspects of ownership, visibility and execution for the associated goal or practice. Overall maturity is then evaluated based on the maturity of the three component aspects. Rather than impose an arbitrary algorithm we make this a subjective process, the only restriction being that the overall rating can't exceed the range set by the individual aspects, hence three wrongs can't make a right! The results for the individual goals or practices are in turn used to guide the overall evaluation of each process area.

All the results are captured in a single easily accessible spread sheet and can be made even more visible through the use of spider graphs to present the key results. This is illustrated in Figure 3 which shows the top level results from an actual project. This visual representation facilitates both reporting the results to management and also presenting results back to the project teams.

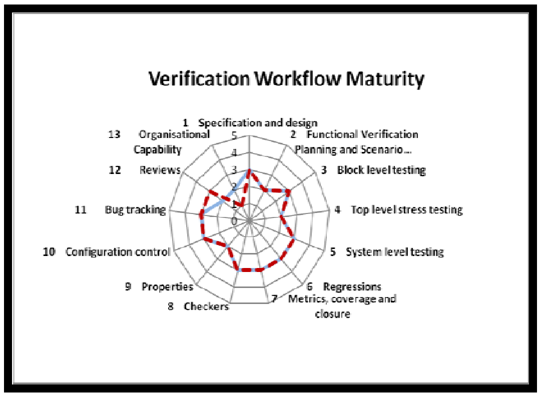

Figure 3: Presenting Verification Workflow Maturity

This process serves to build a picture of the verification workflow maturity. This may be across projects, teams or sites. By repeating the review at regular intervals it is also possible to build a picture over time that can be used to track the impact of process improvement or organisational change. It can also be used to find if there is a mismatch in perception between various team members, or between engineers and management. This can be identified by following a 360 feedback process where members of staff also score the maturity of the different process areas. For example Figure 4 shows results from a project where the less experienced staff were over-optimistic, especially about their own capabilities.

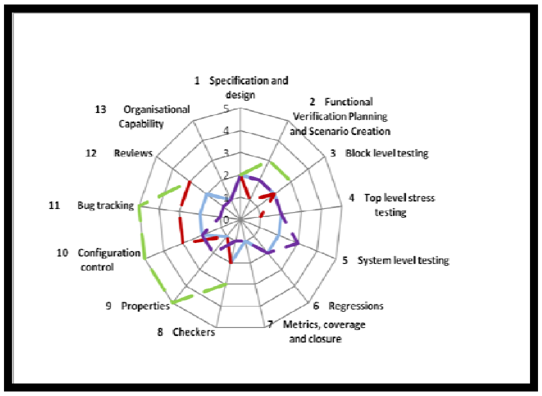

Figure 4 Presenting Results of 360 Feedback Process

The results of the benchmark can also serve to identify both local weak spots and common mode failures that run across projects and sites.

Whilst this evaluation is partially subjective the evidence based 'bottom up' flow aims to ensure the conclusions are fact based and the results can be validated in a number of ways:

1. By comparing to quantitative metrics. Where these exist they can provide objective evidence that is independent of any reviewer bias.
2. By comparing the results from experience external reviewers with self assessment by the verification manager and other experienced verification staff. Our experience suggests that these are often closely aligned, as in Figure 3 that shows such results from an actual project review.
3. By challenging the results when they are fed back to the project teams and to management. Unexpected or counter-intuitive results can be investigated by drilling back down to the underlying evidence.

By defining target maturity levels appropriate to the business and its future product roadmap a gap analysis can be conducted. The results can then be used to identify key issues and plan improvements in either specific processes or in overall functional verification maturity. Regular reviews against this model can ensure the organisation maintains an appropriate level or help drive a process of continuous improvement, though subsequent audits should aim to apply common standards for evaluating maturity.

Summary

Benchmarking is an essential tool, not only for addressing current problems but also helping organisations meet the continually increasing challenges of verification. They achieve this by providing:

1. An objective fact based, view of strengths and weakness
2. A framework for setting goals and priorities, and measuring progress.

Some parts of functional verification can be measured with quantitative metrics, others are subjective. However it is always possible to have a fact based process to classify all capabilities into distinct and meaningful categories.

A practical benchmarking methodology must be easy to customise and lightweight to deploy. This is best achieved by adopting a domain specific solution that avoids some key limitations of a more general framework such as CMMI.

Decomposing capability top down allows a clear link to be made between abstract capabilities and concrete actions. Evaluation can then be performed bottom up using clearly defined maturity levels. The results can be validated by comparing to objective metrics, comparing the results from reviewers with self assessment, and most importantly challenging the results and reviewing the underlying evidence.

Once the results have been reviewed a gap analysis against business requirements helps TVS' customers identify weak areas of their verification process in a timely fashion. Thereafter the FV-CMM provides a framework for management to plan and track improvements.

Acknowledgement

The authors are grateful to our customers who worked with us on the development and deployment of the ideas described in this paper. We would also like to thank the referees for their very constructive feedback.

Reference

Carnegie Mellon University Software Engineering Institute. (2010). *CMMI For Development V1.3.*

Foster, H., & Warner, M. (6/2009). Evolving Verification Capabilities. *Verification Horizons*

Meeth, S. (2010). NVIDIA Formal Verification Metrics and the ITRS Roadmap. *Jasper Users Group.*