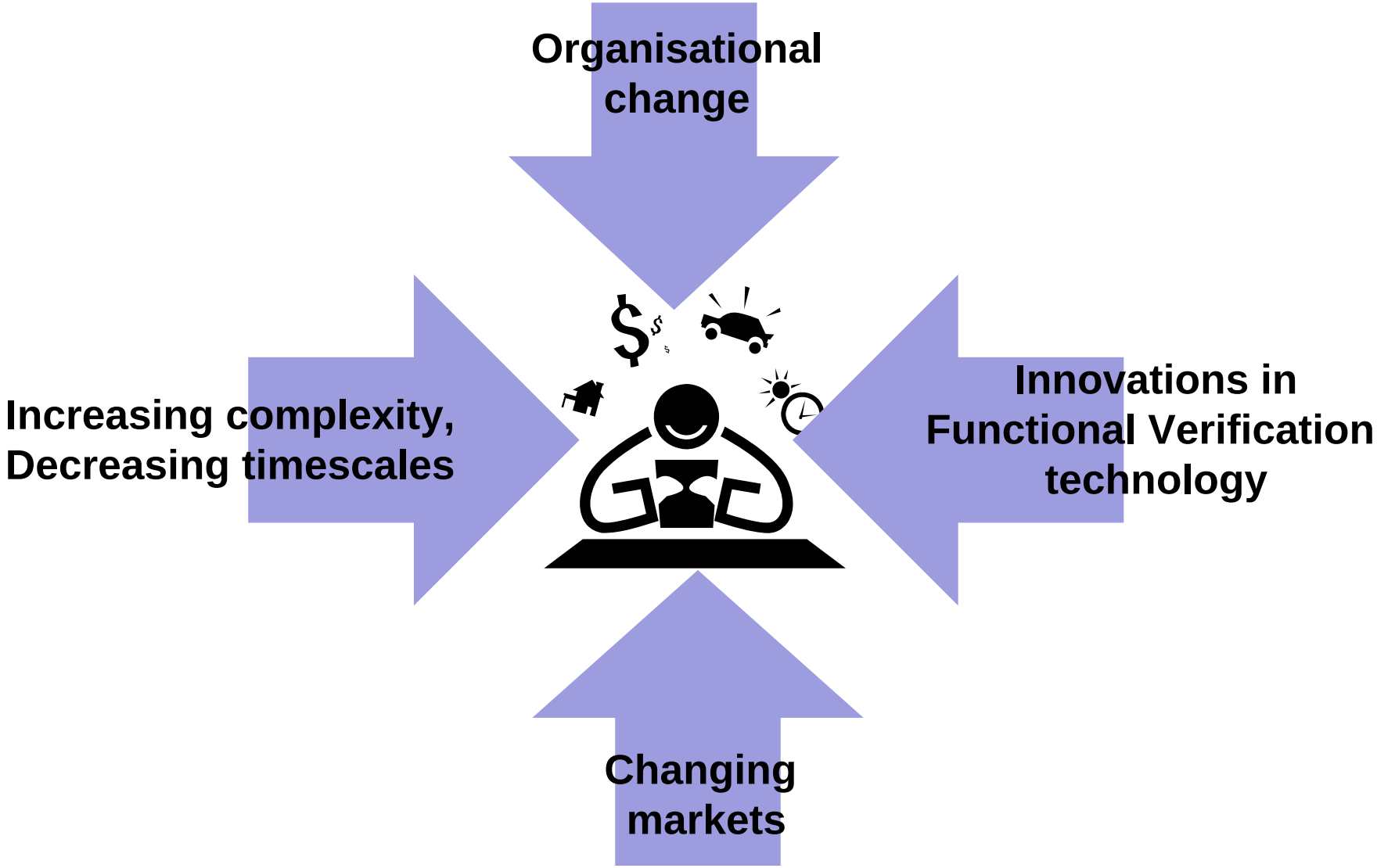
Test and Verification Solutions

Getting you to market sooner by providing easy access to outsourcing solutions

Benchmarking Functional Verification

Mike Benjamin, TVS

Benchmarking - Motivation



Benchmarking - Objective

- A means of measuring the maturity of functional verification activities
- A framework for process improvement that can help management:
  - define goals and priorities
  - measure progress
- A solution that does not **burden** the organisation …especially active projects!

> *tr.v.* **bur·dened**, **bur·den·ing**, **bur·dens**
> **1.** To weigh down; oppress.
> **2.** To load or overload.

Benchmarking - Existing solutions

- **ISO 9000**
  - Aimed at checking adherence to existing practices rather than process improvement
- **Measure against known metrics or roadmaps**

  eg: International technology Roadmap for Semiconductors (ITRS)
  - Allows point by point comparison but tend to lose the ‘big picture’
- **CMMI**
  - Excessive gap between framework and domain specific application
- **Functional verification specific benchmarks**

  eg: ‘The Evolving Capabilities Model’ from Mentor Graphics
  - Methodology should make the link between a high level view of capability and a detailed view of specific verification activities

FV-CMM – Process Areas

1 Specification and design
2 Functional Verification Planning
3 Block level
4 Top level stress testing
5 System level
6 Regressions
7 Metrics, coverage and closure
8 Checkers and properties
9 Configuration control
10 Debug
11 Bugs
12 Reviews
13 Organisational Capability

- Process areas not abstract but specific to functional verification
- Two key elements to capability:
  - METHODOLOGY = HOW
  - PROCESS = VISIBILITY and CONTROLLABILITY
- Includes '*specification and design*', the bedrock of functional verification
- Includes the '*organisational capability*' to learn and adapt

FV-CMM - Structure

- **Process areas**

  = Key activities

  "Specification and design"

  "Reviews"

- **Goals and Practices**

  = Requirements (the "what")

  "*Ensure integrity of code base*"

  "*All tasks should have agreed completion dates*"

- **Actions and Activities**

  = Specific instances (the "how")

  "Minimise configurability of the design and ensure it is made explicit and as orthogonal as possible"

  "*Regression testing, using appropriate scenarios and checkers, is used to validate bug fixes and ensure errors are never reintroduced.*"

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 is 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.

FV-CMM - Evaluation

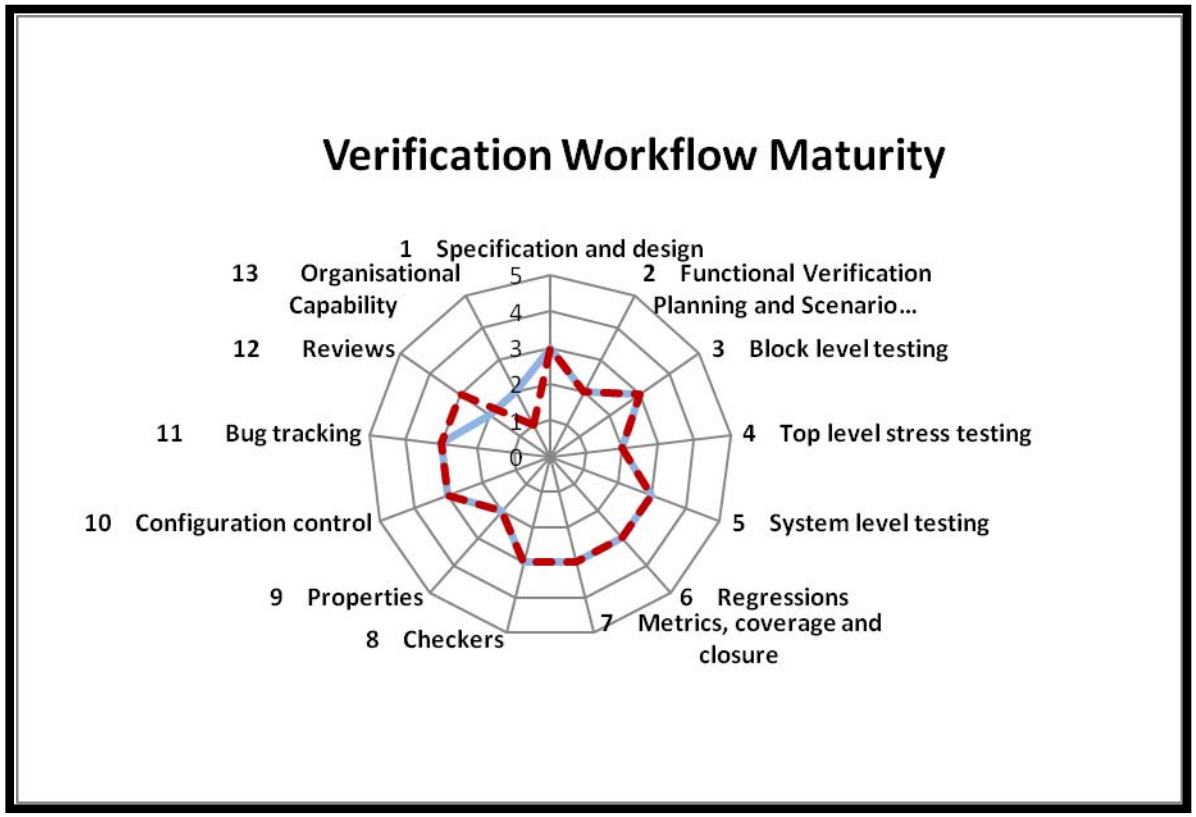
	Initial	Managed	Defined	Quantitative	Optimising
Ownership	Individual	Project Team	Project Stakeholders or ad hoc groups of projects	Community	Company wide or institutionalised
Visibility	Undocumented. No reviews. No metrics.	Documents incomplete or unmaintained. Point reviews. Progress metrics.	Maintained 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.

FV-CMM - Process

- **Customise framework**
  - Process areas are largely fixed but no "*one size fits all*"
  - The specific "**actions and activities**" are often organisation, time or even project specific
  - All captured in a single spreadsheet
- **Interview selected project staff**
  - Interview key staff in small groups (about 3 Engineers)
  - Interviewers work in pairs (interviewer & recorder)
- **Evaluate bottom up**
  - **ACTIONS → GOALS → PROCESS AREAS**
  - Use evidence from interview to score the **maturity** of **ownership**, **visibility** and **execution**
  - Evaluation is subjective with no fixed evaluation methodology but can later drill back down to the evidence

FV-CMM - Applications

- Builds a picture:
  - across projects
    - … or sites
  - over time
  - across a team
- Can identify:
  - common issues
  - local weak spots
    - .. then drill down into the data!



- Validate by …
  - Comparing to objective metrics (where these exist!)
  - Compare results from reviewers with self assessment
  - Challenging the results!

Conclusions

- Benchmarking is an essential tool for helping organisations meet the challenges of the next five years
  - Provides an objective, fact based, view of strengths and weaknesses
  - Provides a framework for setting goals and priorities, and measuring progress
- Some parts of functional verification can be measured with quantitative metrics, others are subjective. It is still 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

  … such as the TVS assureMark™ ☺