CMMI® Institute Global Congress Seattle 2015

Top 10 Software Metrics/Measures for every Software Organization

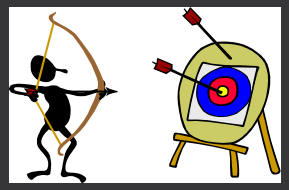
Raghav S. Nandyal

SITARA Technologies Pvt. Ltd.

Chief Executive Officer

raghav_nandyal@SITARATECH.com

SITARA
Excellence Combining Expertise™



OBJECTIVES

- Most software measurement programs get stuck while answering the question: *"which software measure/metric should we collect?"*
  - Popular approaches based on Goal-Question-Metric are more relevant when processes have matured at least to a level 3 or Defined Maturity
- How does an organization that is just getting started with a measurement program, which has no idea of which metric to collect begin the journey of software measurement?
  - Some of the proven software measures which have been found useful and known to add value will be described
- What are the top 10 Software Metrics/Measures which every Software Professional must care about?

Some Perspectives …

(1 of 3)

When organizations embark on a software measurement program

- If a list of measures that they ought to consider and the general value one can expect from using these measures is provided,
- there is a better chance for the measurement program to succeed
- because, individuals working on projects would be geared intrinsically, to offer inputs with which to derive management insights

It is often the case that, organizations at lower maturity levels tend to defer establishing a well reasoned measurement program that can drive business performance until they reach higher maturity levels

- Metrics collection and analysis that help to obtain objective understanding of the state of software practice in any organization must begin early

Some Perspectives …

(2 of 3)

Organizations that delay making measurement and analysis effective until they have mastered the process may miss out on many intuitive opportunities their improvement effort is likely to provide

In order to comply with the CMMI framework, frivolous measures are often collected at lower maturity levels that might not demonstrate improved business results as a natural outcome of effective process institutionalization

- Since measures were frivolous to begin with, process improvement programs downplay the significance of measurement and analysis, and unfortunately do not derive any value from their analysis
- Senior management too, does not see value in measurement and analysis beyond the fact that a lot of time is spent, … there is further dilution of its importance and the vicious circle continues!

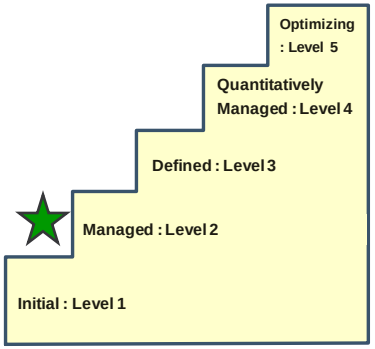
Some Perspectives …

(3 of 3)

Soon, the measurement program becomes an exercise akin to counting the quills on a Porcupine – painful!

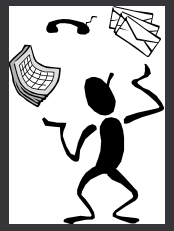
This paper presents 10 software measures that have practical significance *with or without* the aid of a Goal-Question-Metric paradigm, to serve as key process performance objectives

Advantages in knowing what some of these beneficial measures are, helps measurement and analysis, twofold



- One: Process Quantification, an expensive activity might have significantly less rework at L3 and,
- Two: Process Performance Management, a high maturity phenomenon at Levels 4 and 5 can be seamlessly tried after achieving Level 3 process maturity

What are the 10 software measures that have practical significance?

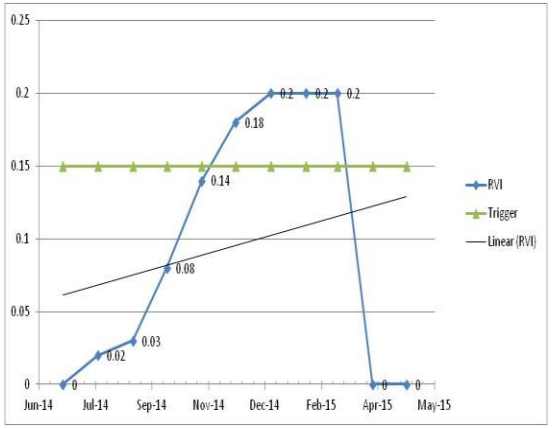


What are the top 10 software measures?

Project Management Metric *(1 of 10)*

Changing requirements often impact most projects

- Changes can be due to internal or, external reasons
- Requirements Volatility establishes a meaningful basis to understand stability of project requirements and understand the causes for “requirements churn”;
- More relevant in early stages of the project (what versus how)
- When this metric is designed and used correctly, it offers insights into how well a project can establish plans for timely completion of project activities



- Example: Requirements Volatility Index [RVI] can be defined as:

  RVI = [ΔMAD R + δMAD R]/R as a percentage

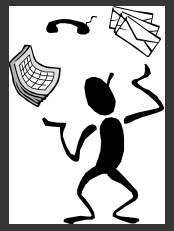
  R: *Original requirements at the time of first baseline*

  Δ MAD R: *Major changes to requirements identified as Modified, Added, Deleted*

  δ MAD R: *Minor changes to requirements identified as Modified, Added, Deleted*

IMPORTANT NOTE: Make sure that there are no double counts.

- RVI is typically used to trigger a management decision process for when re-estimation of Project Planning parameters is required

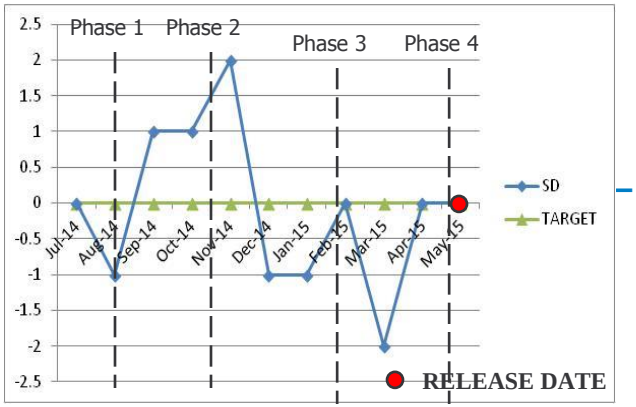


What are the top 10 software measures?

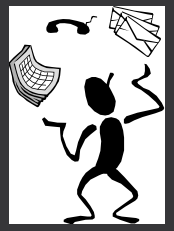
Project Management Metric *(2 of 10)*

How are projects affected by schedule related issues?

- Schedule Performance Index [SPI] establishes an understanding for Schedule Estimation Accuracy
- SPI is understood as a ratio of the planned or estimated duration in days/weeks/months to the actual duration for completing a set of tasks within the critical path or for a milestone achievement
- Or, SPI = Earned Value/Planned Value for "all tasks"



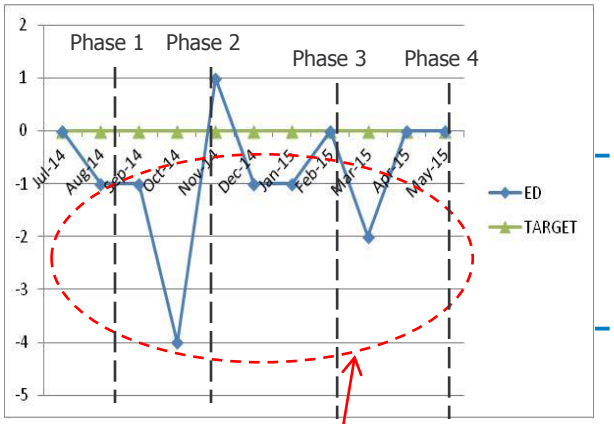
- Phase-wise Schedule Deviation [SD] is expressed as the deviation in days/weeks/months from the planned or expected baseline date for all critical path tasks or phases in the work-breakdown structure
- Allowable slippage or deviation in days/weeks/months is typically set as the "target" based on project or contract characteristics (time and material, maintenance, fixed bid)



What are the top 10 software measures?

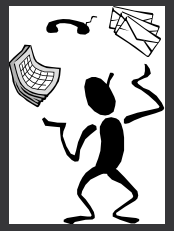
Project Management Metric *(3 of 10)*

How are projects performing on effort estimations?



Pessimistic Effort Estimates: Calls for a review of effort estimation process in use

- Phase-wise Effort Deviation [ED] expressed as the deviation in staff-days/weeks/months from the planned or expected phase-wise effort
- High deviation is indicative of incorrect effort estimation, or, the assigned effort is not meeting “expected” achievement of workload
- Review of staffing effectiveness, balancing the resource-mix are typically undertaken when the phase-wise effort deviation is above a defined threshold or target
- Linked to organizational business objective of “profitability and on-time performance”



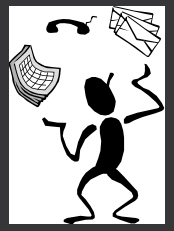
What are the top 10 software measures?

Project Management Metric *(4 of 10)*

How cost effective/profitable is the project vis-à-vis budgeted costs?

- Phase-wise Cost Performance Index [CPI] ratio of budgeted cost to the actual cost required to complete a phase
- Cost overruns or under runs are detected with this metric at the completion of each phase
- Remedial steps can be taken as appropriate, to ensure on-budget or profitable completion of projects prior to a tollgate review

- Other possible ways of representing CPI are possible; for instance…
- CPI can be understood in terms of resource mix, resource loading, contracted v/s planned effort, competency index and
- Process Performance Model can be established using a regression equation as:
  - CPI = 0.75 + 0.945*RM + 0.98*RL + 0.857*CvPE + 0.889*CI

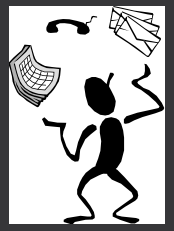


What are the top 10 software measures?

Process Management Metric *(5 of 10)*

What is the hidden-factory?

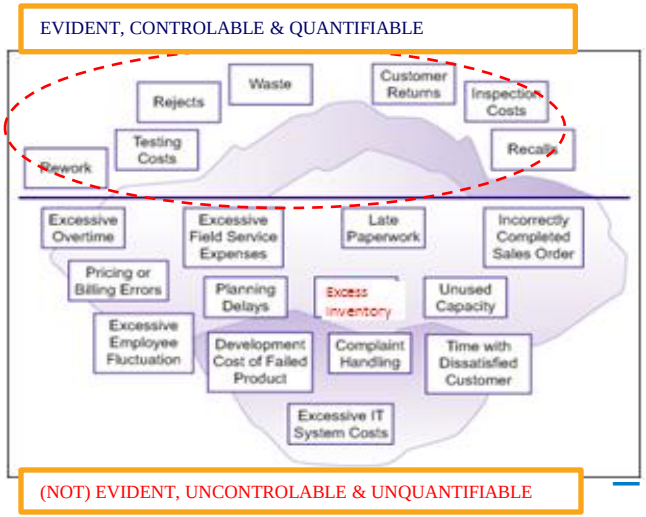
- Cost of Poor Quality [COPQ] percentage of the ratio of the sum of effort/cost required to fix both internal and external failures of a baseline (more simply: all rework effort/cost), to the total project effort/cost at the completion of each phase
- Phase wise tracking of COPQ establishes sound basis for *phase-wise root-cause analysis and resolution;* more focused improvements to eliminate root causes of variation at the (sub-process) level, where it is best controlled, can be conceived
- Other methods of establishing COPQ: multiply the number of non-conformances in a phase with average cost to fix a failure type associated with that phase



What are the top 10 software measures?

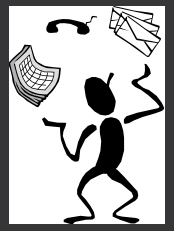
Process Management Metric *(6 of 10)*

How much effort is required by projects to confirm quality of work products being created?



Adapted from: www.isixsigma.com

- Cost of Quality [COQ] percentage of the ratio of the sum of
  - Effort/cost of Poor Quality,
  - effort/cost required to prevent non-conformance,
  - effort/cost to appraise conformance to the stated and implied requirements prior to establishing a baseline
  - to the total project effort at the completion of a phase
- Phase wise tracking of COQ establishes *phase-wise causal analysis and resolution* triggers, to enact more focused improvements to eliminate root causes of variation at the sub-process level



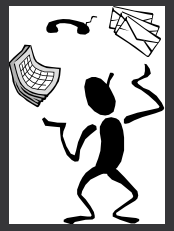
What are the top 10 software measures?

Engineering Metric *(7 of 10)*

What is the screening effectiveness of code reviews/inspections?

- Code Review Effectiveness [CRE] expressed in percentage is the ratio of errors caught during code reviews/walkthroughs and the total errors in code reviews/walkthroughs and defects caught in unit testing after achieving 100% line coverage
  - DEFINITION: Escaped error = Defect
  - DEFINITION: Fault = Error + Defect

- CRE is especially useful in understanding the effectiveness of the coding phase inspection process in relationship to the defects caught by unit test cases used to exercise the structure of code as in a typical white-box testing activity
- It offers a sense for structural correctness besides algorithmic accuracy of the code or logic (something which cannot be ascertained using automation tools/scripts/solutions only)

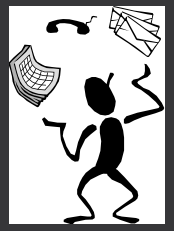


What are the top 10 software measures?

Engineering Metric *(8 of 10)*

What are the defect-prone tendencies in a phase causing defects in work-product?

- Phase-wise Defect Density [DD] ratio of number of defects occurring in a phase to the work-product size of the baseline produced at the end of each phase
- A lag indicator, phase-wise defect density helps to understand strengths/weaknesses of internal competencies and of the mistake-prone tendencies while executing a process
- Defects in a phase or a work-product produced is established at the completion of the project; corrective and prevention actions for the next cycle or in subsequent project execution are identified
- During postmortems, defects are attributed to the phase in which they should have ideally been prevented from occurring
- This allocated number is divided into the work-product size defining the baseline characteristic (E.g., defects/KAELOC)



What are the top 10 software measures?

Engineering Metric *(9 of 10)*

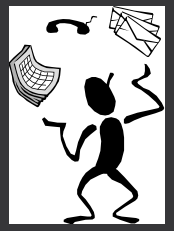
What is the screening effectiveness of reviews in a phase?

- Phase Containment Effectiveness [PCE] ratio of errors detected in a phase to the total faults (errors and defects) assigned to a phase
  - Again, DEFINITION: Escaped error = Defect
  - … DEFINITION: Fault = Error + Defect

- A lag indicator, phase containment effectiveness helps to understand the strength of reviews or screening mechanisms while executing a process prior to forming a baseline

PCE is defined for a phase as,

$$(PCE)_{Phase} = \frac{(Errors)_{Phase}}{(Errors + Defects)_{Phase}}$$



What are the top 10 software measures?

Support Metric *(10 of 10)*

How does effort consumption compare to work accomplished on fixed-bid projects?

- Effort to Time Ratio [ETR] Deviation: offers useful insights into the project completion status at milestone/important junctures and effort utilization and
- therefore serves as a lead indicator of project progress
- Typically, a +/-10% deviation is set as a trigger to establish control and oversight into why either the effort or the elapsed time (schedule) is more than normal
- ETR indirectly establishes the accuracy of the estimation and effectiveness of the staffing process/resource allocation
- ETR is based on two key measures which are directly correlated - effort utilized and the elapsed cycle time of the project
- When projects involve a fixed staffing profile, expended effort and the elapsed cycle time of a project must be close to 100% if the effort estimation process used and the resource estimates are accurate

Conclusion

(1 of 3)

These definitions are “basic”; they can always be sharpened and improved to meet the demands of the different business/organizational circumstances

Good News: collection of very fundamental project/process measures (effort, schedule, errors, defects) is all it takes to establish these derived performance parameters

Better News: start out with these, and gradually improve upon them; if there is an error initially, all projects will be offset to a similar extent; being exact with software metrics is never the expectation!

Great News: you know how to get started if you have not already, or continue to improve the state of your measurement practices

Conclusion

(2 of 3)

Bad News: No single metric by itself is useful! They have to be read and understood together in some meaningful combination

When these performance measures are consciously linked to business or organizational goals/objectives, a natural transition to high process maturity happens

- Building high process maturity is the easy part; sustaining the gains resulting from high process maturity is the more difficult aspect of process management
- Having witnessed fantastic benefits resulting from high process maturity over the last 25 years, one thing is for certain: *the "fun" element in establishing the improvement program is lost if "all" stakeholders are not fully involved/committed to measurement*
- Measurement and analysis can never add value if "all" stakeholders don't *accurately* do whatever it takes in knowing how their contribution or lack of it, affects project performance

Conclusion

(3 of 3)

- Example:
  - Business Objective: Improve Quality of Internal and External Deliverables prior to UAT

PCE for Requirements Phase

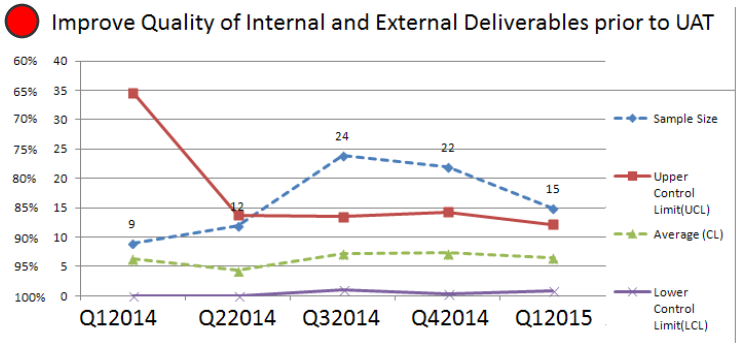
PCE for Analysis and Detailed Design Phase

PCE for Coding Phase

PCE for Unit and Integration Testing Phase

PCE for System Testing Phase

Select homogenous samples for trend analysis

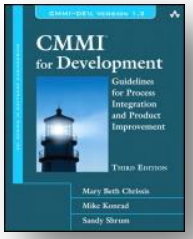


A related paper is being presented by SITARA in this conference titled:

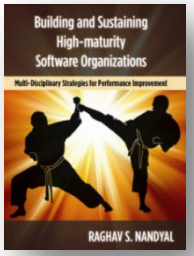
"Key Features of a Good Process Capability Baseline Report"

I invite you to attend the paper presentation/review the slides in the proceedings of the CMMI Institute Global Congress Seattle 2015

References



[Chrissis 2011] Chrissis, Mary Beth et al, “CMMI for Development®: Guidelines for Process Integration and Product Improvement, Third Edition”, Addison-Wesley Professional. 2011



[Nandyal 2012] Nandyal, Raghav, “Building and Sustaining High-maturity Software Organizations”, McGraw-Hill Education. 2012

[Nandyal 2012] Nandyal, Raghav, “How to Accelerate High-Maturity Implementation of CMMI using Six Sigma”, Proceedings of the 24th SEPG 2012. Albuquerque, NM USA

[Nandyal 2015] Nandyal, Raghav, “Key Features of a Good Process Capability Baseline Report”, Proceedings of the CMMI Institute Global Congress 2015. Seattle, WA USA

Thank You!

DISCUSSIONS, Q&A

Raghav S. Nandyal
Chief Executive Officer

SITARA Technologies Pvt. Ltd.
#54, Sri Hari Krupa
6th Main Road
Malleswaram
Bangalore KA 560 003 INDIA

SITARA Technologies
435, Marshall Road
Southlake, TX 76092
USA

Telephone: +91.80.2334-3222
Mobile: +91.984-523-3222
Email: raghav_nandyal@SITARATECH.com
URL: http://www.SITARATECH.com