

V&V Metrics

V&V Metrics

Primary questions:

- Where have I been?
- Where am I going?
- When will I get there?

Management questions:

- Where have I been? (as above)
- Where am I going? (as above)
- When is it time to stop?

Verification metrics: When is enough enough? <https://www.edn.com/electronics-news/4326023/Verification-metrics-When-is-enough-enough->

Decision

“When is it time to stop? “ depends on

- Fusion of diverse metrics
- Detailed V&V plan that existed from the beginning and has grown and blossomed along with the design.

Verification metrics: When is enough enough?

<https://www.edn.com/electronics-news/4326023/Verification-metrics-When-is-enough-enough->

Definitions

- Measure – Quantitative indication of the extent, amount, dimension, capacity or size of some attribute of a product or process
- Metric – Quantitative measure of the degree to which a system, component, or process possesses a given attribute
 - “Often metrics are ratios – bugs/developer, actual project duration/planned project duration
- Indicator – Metric or combination of metrics that provide insight into the software process, a software project or the product itself

Software Metrics:

<https://www.slideshare.net/swatisinghal/software-metrics-5079475>

Indicators

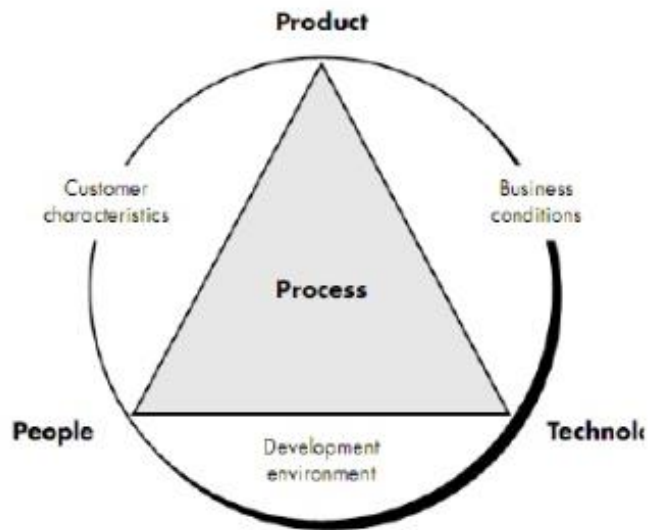
Only 4 gauges are critical to fly a plane: heading, airspeed, altitude and attitude (pitch and yaw). Others are for communication, maintenance, troubleshooting, etc.

Choose the metrics you'll watch carefully.



Only a few key instruments out of the entire dashboard are needed for basic flight.

Types of Metrics



1. Process metrics
2. Product metrics
3. Project metrics

Software Metrics:

<https://www.slideshare.net/swatisinghal/software-metrics-5079475>

Metrics

- Code coverage – automatic
 - line coverage
 - branch coverage
 - path coverage
- Functional coverage - how many of the functions of the design you have shown to do what you intended
- Constrained random tests
- Assertions & assertion coverage
- Property checking tools (formal tools)
- Frequency of bug reports

Verification metrics: When is enough enough? <https://www.edn.com/electronics-news/4326023/Verification-metrics-When-is-enough-enough->

Fusion of metrics

Possibility:

- Start with functional-coverage, as get 100%
- Turn on code line-coverage as a check of completeness
- Frequency of bug reports – ought to be decreasing as coverage tests near 100%
- Watch critical blocks – new functions, blocks with which the designers lack experience

Verification metrics: When is enough enough? <https://www.edn.com/electronics-news/4326023/Verification-metrics-When-is-enough-enough->

Metrics/Measures

- Lines of code
- Number of classes & interfaces
- Code to comment ratio
- Cyclomatic complexity – number of linearly independent paths through source code
- Code coverage
- Bugs to lines of code ratio
- Cohesion (high, med, low)
- Coupling (high, med, low)
- Failed tests per build
- Version control commits per day
- Lines of code per commit

Testing related formulas

% Effort Variation = (Actual Effort - Estimated Effort) / (Estimated Effort) * 100

% Duration Variation (Optional) = ((Actual End Date - Actual Start Date) - (Planned End Date - Planned Start Date)) / (Planned End Date - Planned Start Date) * 100

% Schedule Variation = (Actual End Date - Planned End Date) / (Planned End Date - Planned Start) * 100

Load Factor = (Actual Effort / Effort Available)

%Size Variation = (Actual Size - Estimated Size) / (Estimated Size) * 100

Test Case Coverage % = ((Total number of test cases or scripts prepared) / (Total number of test requirements * total number of test scenarios)) * 100

Residual Defects Density = (Total number of Post-Production Defects) / (Size of the Application (or) Requests)

Test effectiveness (in percentage) = ((Total no of application defects found - Total number of application defects are rejected by the customer) / (Total no of application defects found + Total number of defects found by customer during UAT)) * 100

Overall Productivity = (Overall Project Size) / (Total Effort for the Project)

Test Case Preparation Productivity = (No of Test Cases / scripts) / (Effort spent for Test Case/script Preparation)

Test Execution Productivity = (Size for Test Execution) / (Effort for Test Execution)