

Project Metrics

Anything you need to quantify can be measured in some way that is superior to not measuring it at all.

8/26/2023

vijaynathani.github.io

1

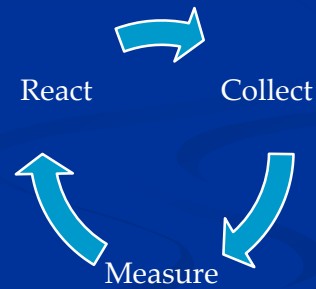
The purpose of Metrics is so that we can decide how to improve the system.
What is measured, gets managed. What is managed, gets done.

Many teams check and adjust based on gut feelings or some idea from Internet.
They don't use real data to determine their direction.

Retrospectives that use data will likely be more productive than using opinions.

Good Metrics

- Comparative
- Actionable
- Accessible
- Auditable



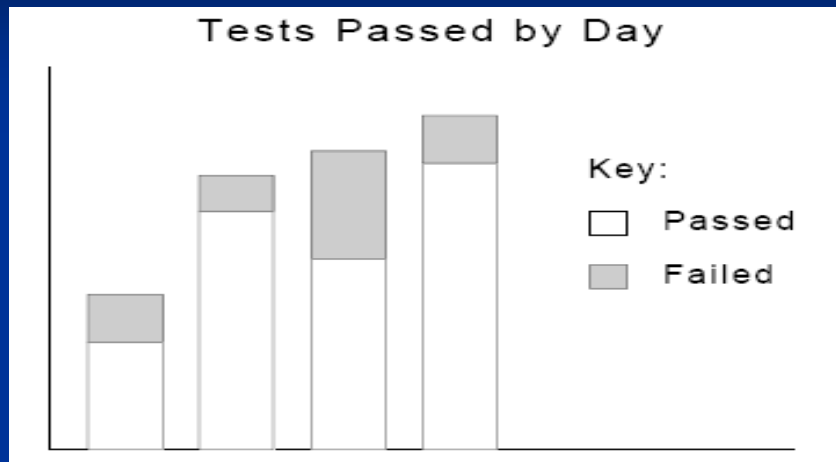
Comparative – Our sales are 1 Million units makes little sense. Our sales are 20% higher than the same period last year is better.

Actionable – A person should know what to do, when he looks at it.

Accessible – It should be available regularly on a frequent basis.

Auditable – Can be verified e.g. A team wanted to know what effect a change will have on call time in call center. The metric is useless if we don't measure the call time currently.

Typical Metric



Sprint Boards



The image shows three types of Scrum cards. The top card is a 'Feature' card with fields for 'Feature ID', 'Business Value', and 'Size (Points)'. The middle card is a 'Story' card with fields for 'Story ID', 'User Story', 'Acceptance Criteria', and 'Status'. The bottom card is a 'Task' card with fields for 'Task ID', 'Task Name', 'Task Description', and 'Status'. To the right of the cards is a vertical column of four colored circles: blue, green, yellow, and red.

Post-It's:

- White- Feature
- Yellow- Story
- Blue- Task

Dots:

- Blue- Task in progress
- Green- Task complete
- Yellow- Dependency
- Red- **Impediment** / Blocked

8/26/2023 3:05 PM

vijaynathani.github.io

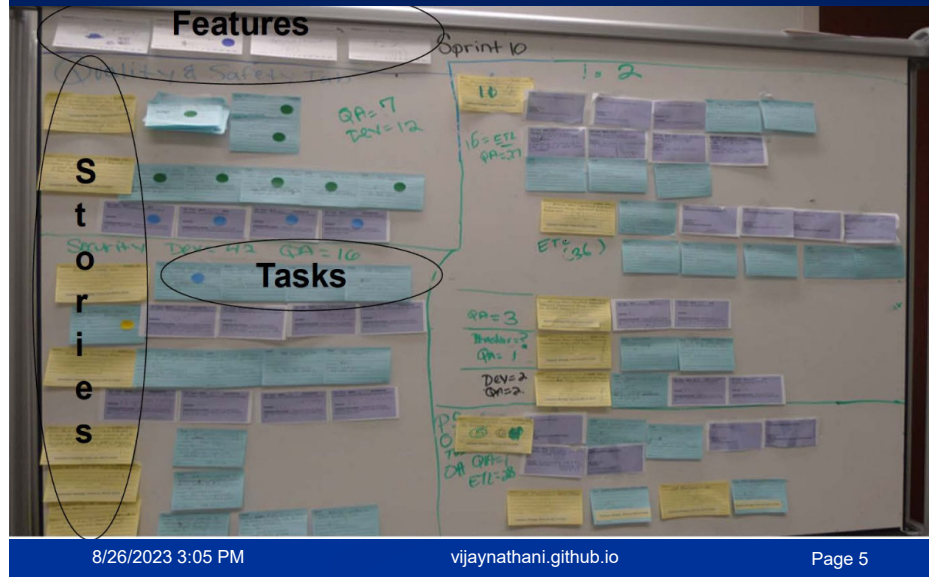
Page 4

White card – Feature

Yellow card – user story

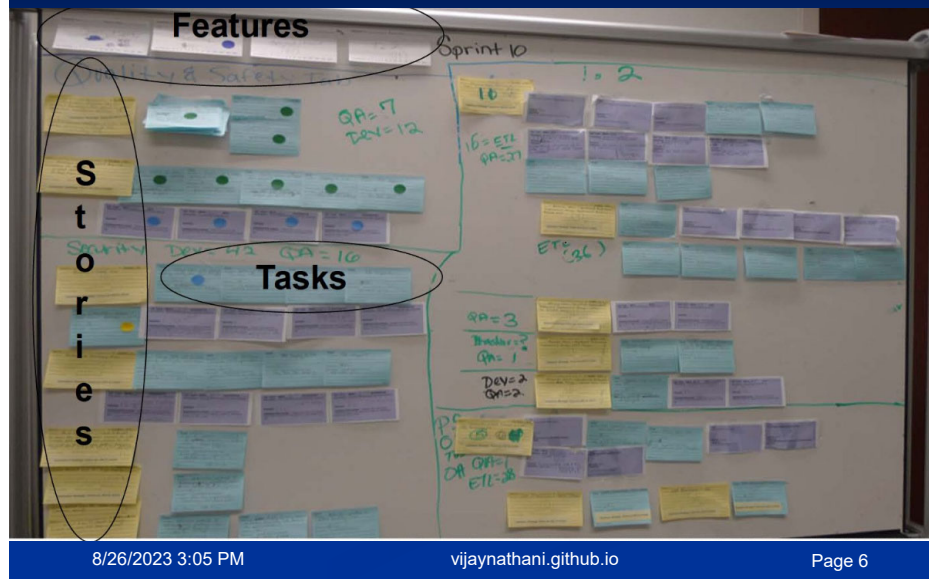
Blue card – Task

Status of Current Iteration



Taken from <http://www.infoq.com/presentations/The-Power-of-Visibility>

Status of Current Iteration



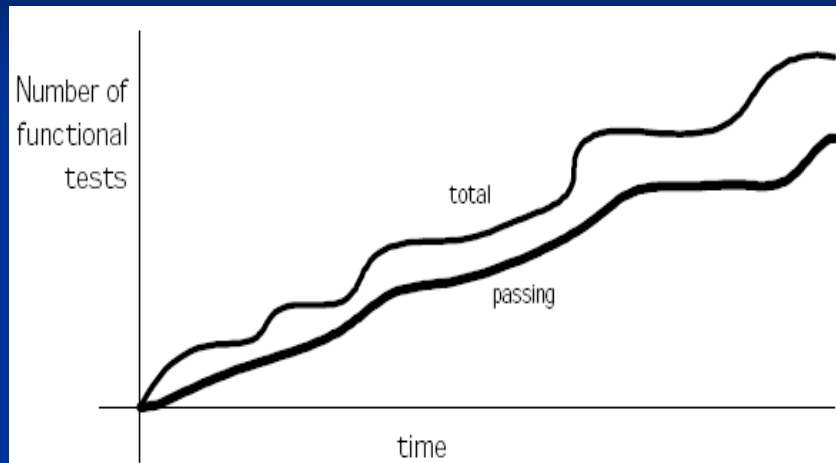
8/26/2023 3:05 PM

vijaynathani.github.io

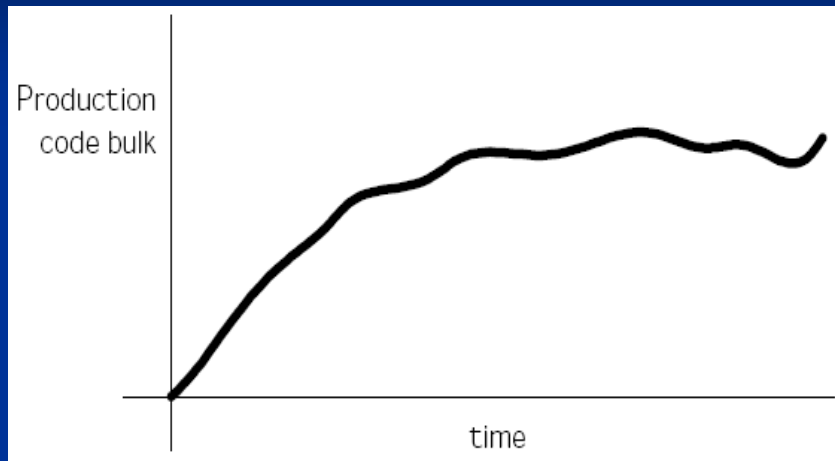
Page 6

Taken from <http://www.infoq.com/presentations/The-Power-of-Visibility>

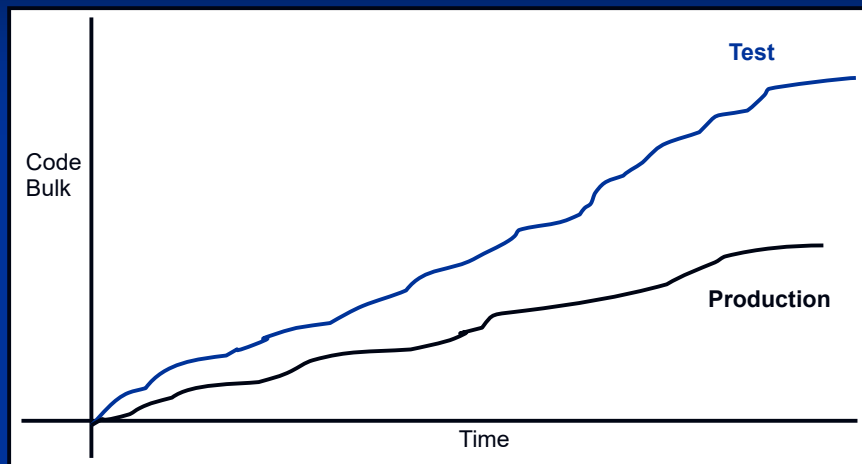
Typical Metric



Typical Metric



Code & Tests LOC



8/26/2023 3:05 PM

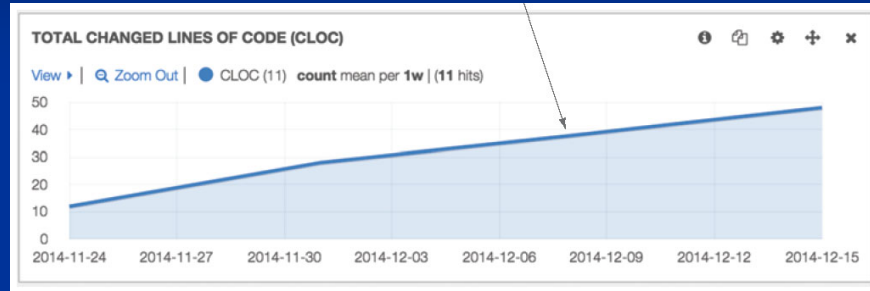
[vijaynathani.github.io](https://github.com/vijaynathani)

Page 9

The size of test code will usually be more than the LOC of the code. If the size of the test code is less than the size of project LOC, then probably the team is not writing enough tests.

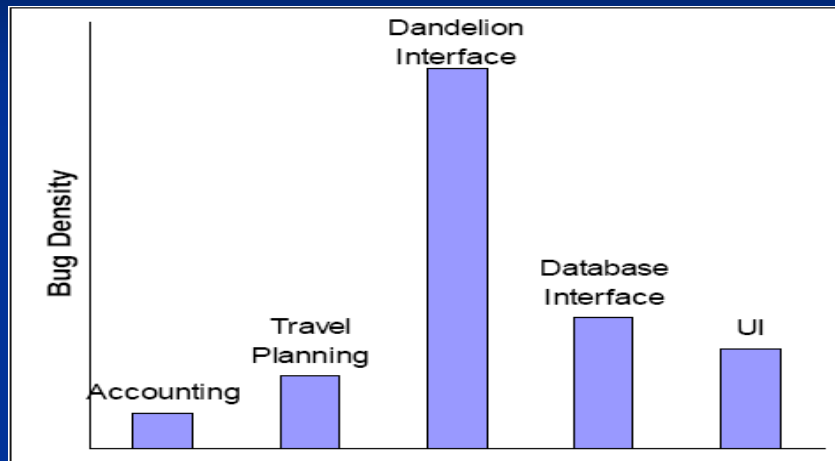
If Production code is 100KLOC, expect the Test LOC to be between 150KLOC to 300KLOC.

CLOC

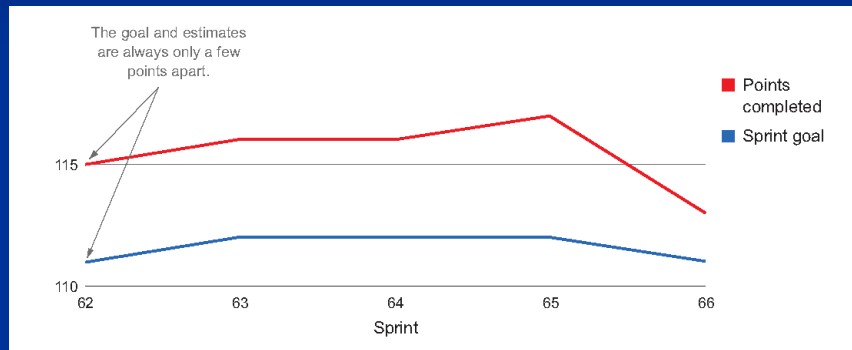


Changed Lines of Code

Cluster for Bugs



Progress by Iteration



8/26/2023 3:05 PM

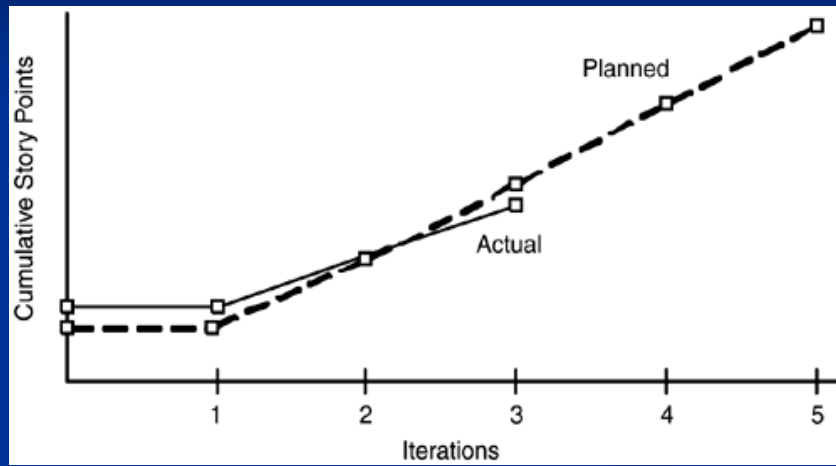
vijaynathani.github.io

Page 12

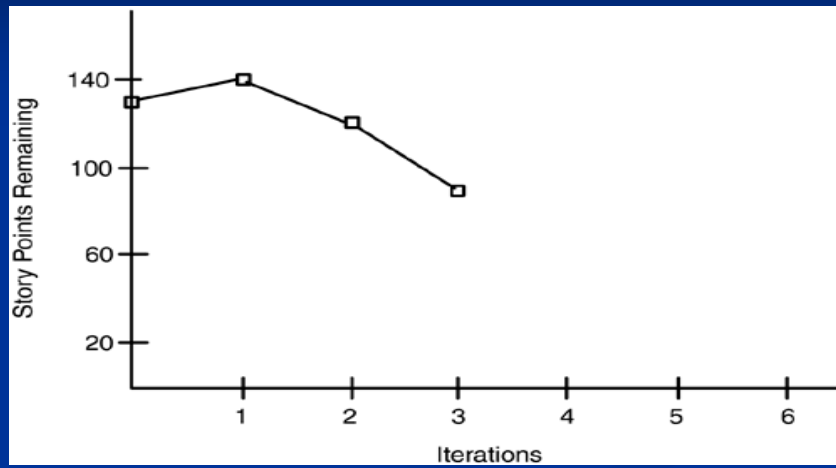
If estimates are this accurate, then team is likely overestimating. Because they have a buffer, they are able to meet their goals.

This tells us about capacity of the team. A Manager, who tries to jack up the velocity by pushing the team, is misusing this metric.

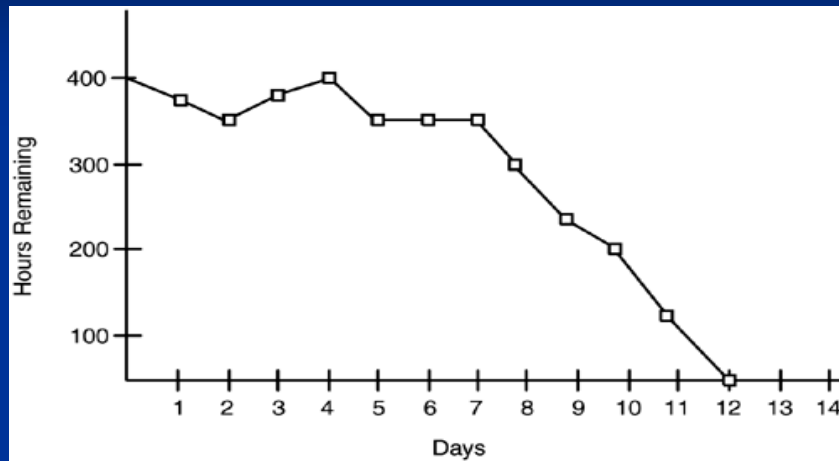
Cumulative Progress



Product Burndown Chart



Sprint Burndown Chart



8/26/2023 3:05 PM

[vjaynathani.github.io](https://github.com/vjaynathani)

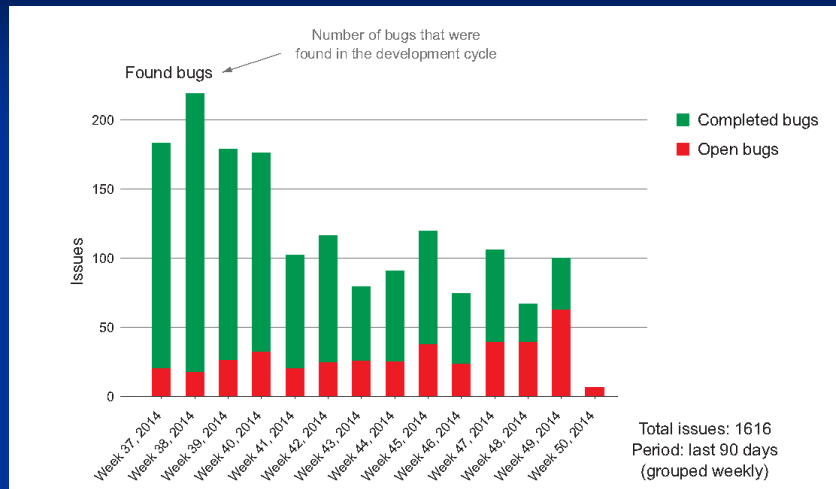
Page 15

To generate these reports automatically, tools exist. However for a small project, paper tracking seems easier.

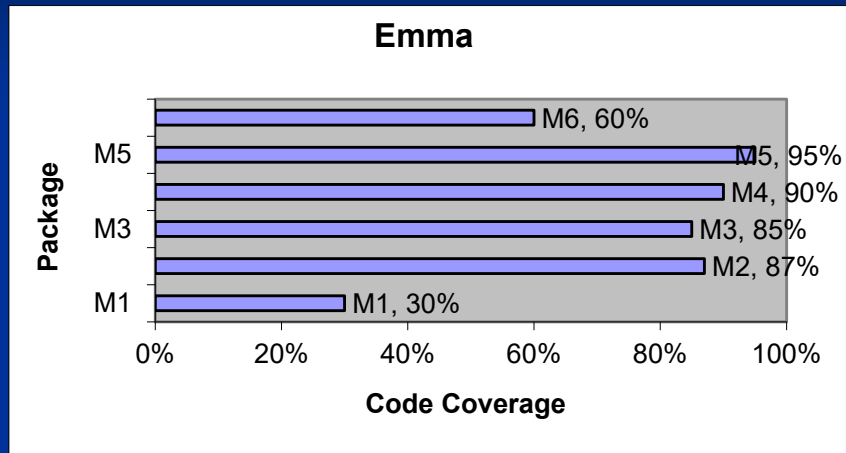
For large teams these tools are useful.

Some of the tools are VersionOne, Rally Software, ScrumWorks, XPlanner.

Bugs

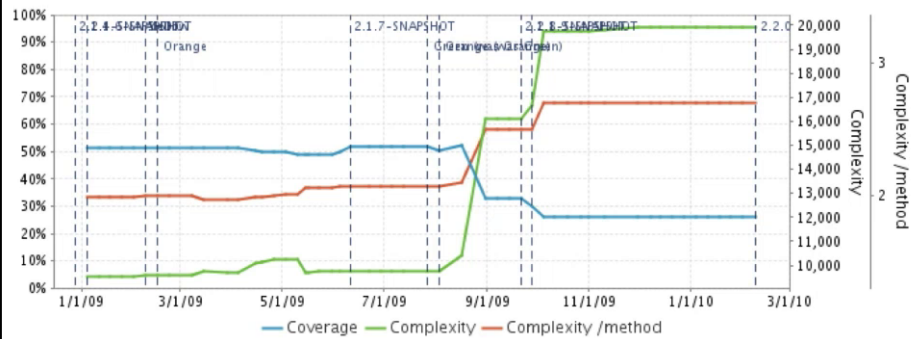


Code Coverage



Code coverage in automated tests using tools like Emma, Cobertura, Clover, etc.

Struts – Time Machine



Mentioned in “Software Architecture Fundamentals” part 2 by O’rielly. Author Neal Ford, Mark Richards.

Something horrible happened on the project on 1-Sep-2009.

Productivity change after Scrum

Months since Scrum implemented	3	12	24
Productivity (product backlog requirements per \$100,000 investment)	4.5	9.0	12.2
Quality (Bugs per release)	100+	100	5

Source: Jeff Sutherland in RootsofScrumJAOO28Sep2005.pdf

Other Metrics

- Tools like PMD to detect potential problems and duplication
- Common errors reported by tools like Findbugs
- Number of defects found after release
- Stability of a package with Jdepend
- Cyclomatic Complexity Number with JavaNCSS

Code coverage of typical agile projects is between 80 to 95%.

Defects reported

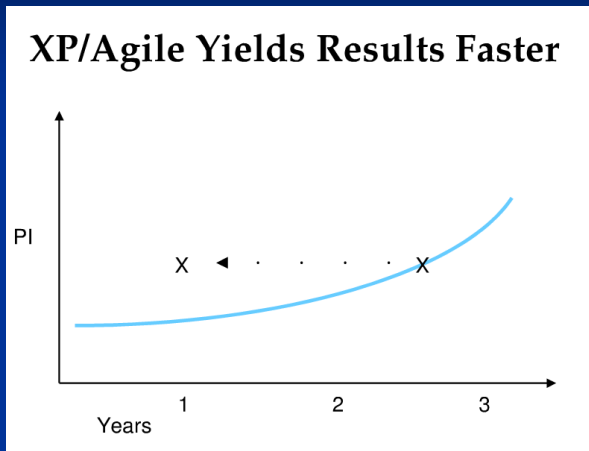
Some XP teams that have been on the path of improvement for several years see only a handful of defects per year.

No defect is acceptable; each is an opportunity for the team to learn and improve.

Metrics for Finance Managers

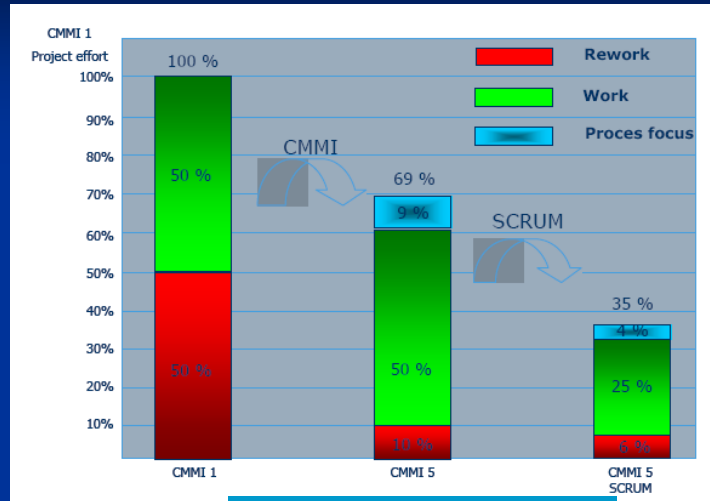
- Revenue / Person
- Profit / Person
- ROI
- Cycle Time - Time lag between the beginning of investment in an idea and when the idea first generates revenue.
- PI (Process Productivity)
 - $PI = (Size\ in\ KLoc) / (Time\ in\ months * Effort\ in\ Person\ Months)$

Example usage of PI Metric



After doing agile, the PI index moved for a company. See details at <http://www.infoq.com/presentations/10-tips-for-agile-transitions>

Productivity at Systematic



Source: ScrumPapers20070424.pdf

Systematic was established in 1985 and employs 371 people worldwide with offices in Denmark, USA and the UK. It is an independent software and systems company focusing on complex and critical IT solutions within information and communication systems. Often these systems are mission critical with high demands on reliability, safety, accuracy and usability.

Customers are typically professional IT-departments in public institutions and large companies and service industries. Systematic was appraised 11 November 2005 method and found to be CMMI level 5 compliant.

This is documented in a presentation on Infoq website. Presentation by Jeff Sutherland.

Process Metrics Guidelines

- Use common sense and organizational sensitivity when interpreting metrics data.
- Don't use metrics to appraise individuals.
 - Metrics are usually not that accurate
- Never use metrics to threaten individuals or teams.
- Metrics data that indicate a problem area should not be considered "negative." These data are merely an indicator for process improvement.

Bad Metrics

- Vanity Metric
 - Makes us feel good/bad about the product
 - Doesn't help us to take decision.

Metrics must demonstrate cause and effect
Not *more* metrics, the *right* metrics

8/26/2023 3:05 PM

vijaynathani.github.io

Page 25

Vanity metics:

- 1) How many people leave comments on our webpage?
- 2) How many people downloaded out app? (not who are using)
- 3) Page Views
- 4) Time on Site

KPI – Key Performance Indicators.

Less is good. Choose one useful metric over 10 useless ones.

- 1) Sign-up, Trail and conversion rates
- 2) Profit per ...

At Amazon, the market VP was against “Customer buying this product also bought ...” He said that once a customer has put something in shopping cart, he should not be distracted. It was a NO, NO, NO.

But the team went ahead and did the experiment. The sales increased by nearly 3% because of this one Idea.

At Microsoft, they found that “Experts are usually wrong”. So we need to measure and then take action.

Some Random Metrics

- Only about 15% of the development effort is in coding.
- 20% of the modules account for 80% of the cost.
- Pair Programming catches 60% of the errors.
- Defect density is average 1-3 defects per hundred lines of production code for projects (most projects considered were without automated tests)
- 20% of the software developers are responsible for 50% of the work done on a project.

Note: The above metrics are only indicative and not absolute.

Not everything that can be counted counts, and not everything that counts can be counted perfectly.

The skill and motivation of the software people doing the work are the most important factors that influence software quality.

Measurement can be seen as a detour. This detour is necessary because mostly are not able to make clear and objective decisions (without quantitative support).

The time to complete a project is not proportional to the number of people working on the project.

Size-oriented metrics are widely used, but debate about their validity and applicability continues.

Summary

Information is the currency of democracy –
Thomas Jefferson

Agile is

- A set of Methods & Skills
- OR
- A leadership mindset & culture

Cultural change is the hardest part of the Scrum – Jeff Sutherland.
A co-operative game – Alistair Cockburn.

References

- Extreme Programming Explained –
Kent Beck, Cynthia Andres
- Enterprise and Scrum – Ken Schwaber
- User Stories Applied – Mike Cohn
- Maverick & Seven day weekend - Ricardo Semler

The End

"Believe nothing, no matter where you read it or who has said it, not even if I have said it, unless it agrees with your own reason and your own common sense."

- Buddha