

Measures & Quality

Measures

- Any process and activity can be improved by using measures to monitor its performance.
- In S/w Engineering, we need to measure products as well as measure development process.
- In agile, we need a constant tracking during entire process of s/w development.
Ex: Team Velocity, Project velocity.
- Every role is responsible for the measure.
- Some measures are presented Daily like daily status within iteration
- Some measure are presented every iteration like code coverage, etc.
- Some measures are presented every release like customer satisfaction
- Compare the development process of s/w to the development process of H/w (road) (Tangible) is a very difficult task.

- Because a professional person can estimate how long it will take to build based on his previous experience.

Ex: How many floors already completed for
How many hours or day.

- While, In S/w it does not work so easily because:

① We can't measure what portion of S/w has been developed so far.

② No clarity on how the measures should be taken. i.e according to what parameters should we measure

- Shall we use no. of lines of code?

- Shall we measure progress by no. of work hours?

- Shall we use no. of tests we developed today or during the last week?

- But still this will not tell us exactly how far we are in the development process and what we still have ahead.

- This is because, the ultimate target of S/w project is always not clear at early stages of Development.

In order to know about measures more clearly, we need to understand the answers for the following questions :

- ① Why are measures needed? நோக்கு
- ② Who decides what is measured? நான்
- ③ What should be measured? நோக்கம் -
- ④ When are the Measures taken? நாள்தேர்
- ⑤ How are the Measures taken? மீது
- ⑥ Who takes the Measures? நான் என்று
- ⑦ How are the measures used? நோக்கு

① Why are Measures needed?

- To control and monitor s/w development processes & products.
- In some projects, measures are not taken at all. Even though the measures are taken, they are not explicitly setting specific goals.
- So, a * set of measures, defined for s/w project should have the following characteristics:

① The measures should be mapped to the project goals

② The measures collection should not affect the process performance or product.

- Measures help the agile sw development team to get a constant feedback from the different components of sw Development environments, people and code.

Ex ① If a measure is people-oriented, then it will measure customer satisfaction, amount of time worked on the product or no. of people worked on the product

Ex ②: If a measure is code-oriented, it will measure the code coverage, level of the code, no. of unit tests, no. of defect

- To conclude, Measures increase Project Transparency

③ Who decides what is measured?

- In agile, Measures are determined by the customer, development team and the organization.

- Each party (role) decides what to measure based on its interests in development process and artefacts.

① Customer is interested in measuring development progress and quality of the artefacts performances.

② Development Team is interested in measuring the impacts of development Methodology, satisfaction of people involved and quality of artefacts from technical perspective.

③ Management People are interested in business aspects like project costs and return on investment, customer satisfaction.

③ What Should Be Measured?

To Increase the productivity

- How many hours per day do teammates work to produce code?
- How many hours per day do teammates work in pairs?
- What is the pairs turnover?
- What is the actual size of development work?

- Teammates need to decide about the best ways to measure the size of their product and what information should be gathered from each pair.
- Set of Measures should be refined and adapted, when needed.
- Measures should be simpler which reflects the actual measurement.
- Only several measures should be chosen, large set of measures can negatively influence

④ When are Measures Taken?

- In agile, we need constant feedback
- In agile, there should be continuous integration, so measures should be measured several times each day.
- In this case, it would be preferable that measures are taken automatically.

Ex:- No. of hours invested each day in development Tasks.

- Hour distribution among tasks during the day.
- Hour distribution among projects during a day.

⑤ How are Measures taken?

- Measures are collected from different role representatives assigned.
- Tracker is responsible for the measure collection & their presentations.
- Tester is responsible for the measures that deal with code quality.
- User evaluator is responsible for measures that reflect user's involvement in the design.

⑥ Who takes the Measures?

- All the team Members involved in Measuring the SW Development process either by reporting the required information to the team members who are responsible for specific Measures.

⑦ How are Measures Used?

- It is not possible to observe measures on regular basis to communicate the project status.
- Measures should be evaluated against project goals.
- During iterations, the goals & subgoals are refined, so the measured which are determined need to change if needed.
 - actual info is accessed to check with current goals.

Monitoring a Large Scale Project by Measures

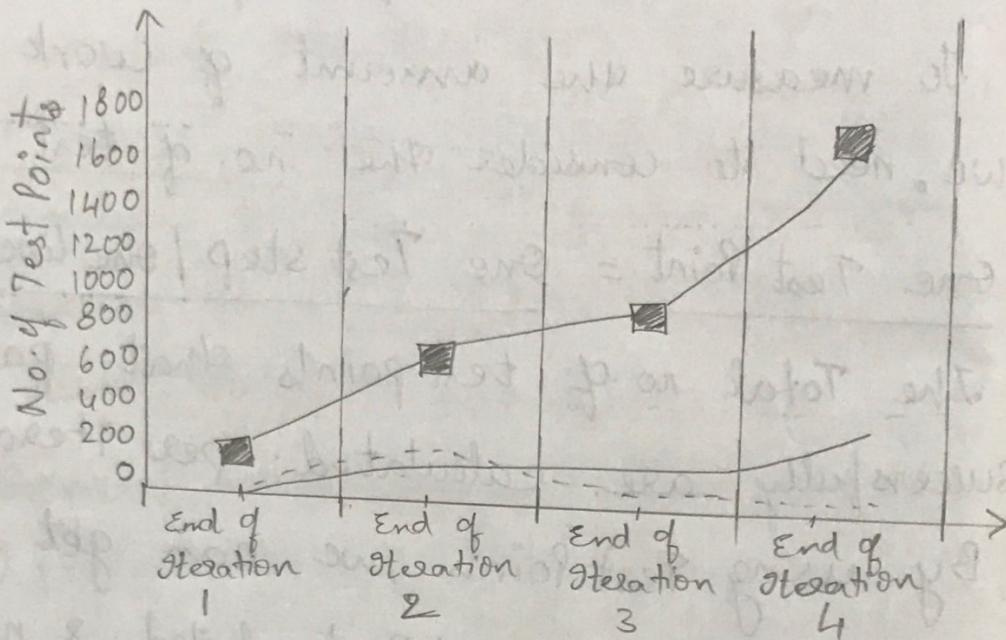
- Initially every project is started with ambivalent feelings (no clarity): The team want to start the development of a prototype.
- The Measure Design will begin at project Risk analysis phase. During this analysis, a measure is added when it seems valuable for risk reduction.
- In General, there are 4 Measures
 - ① Product Size
 - ② Pulse
 - ③ Burn - Down
 - ④ Faults
- These 4 measures reflect the project state, information about the total work, quantity of work performed so far (Product size), speed of the work progress (Pulse), the status of remaining work vs the remaining human resources (Burn-Down) and the overall faults/Errors of the project (Faults).
- Here we consider an example to explain about all 4 measures. We take a four 2-week iterations of a project.

① Product size

- The aim of product size measure is to present the amount of completed work.
- To measure the amount of work completed, we need to consider the no. of test Points.
- One Test Point = One Test step / one line unit Test
- The Total no. of test points that pass successfully are calculated per iteration.
- By using Test Points, we can get the no. of Tests pass, no. of Tests failed, & no. of Tests that did not run at all.
- Test Points are the only measure to measure the Project Productivity. (It's not the no. of lines of code or don't the no. of working hrs)
- The risk involved with Test Points : Some of the test points may not be covered or developer may not write all possible tests. But in this scenario, By seeing the previous measure result, he may change his behavior and write more no. of tests. (If measures not satisfactory in his work for previous Iteration) so he will work more for next iteration.

Example: Take one big project, and consider 4 ~~2 week~~ iterations of it, where each iteration will be for 2 weeks.

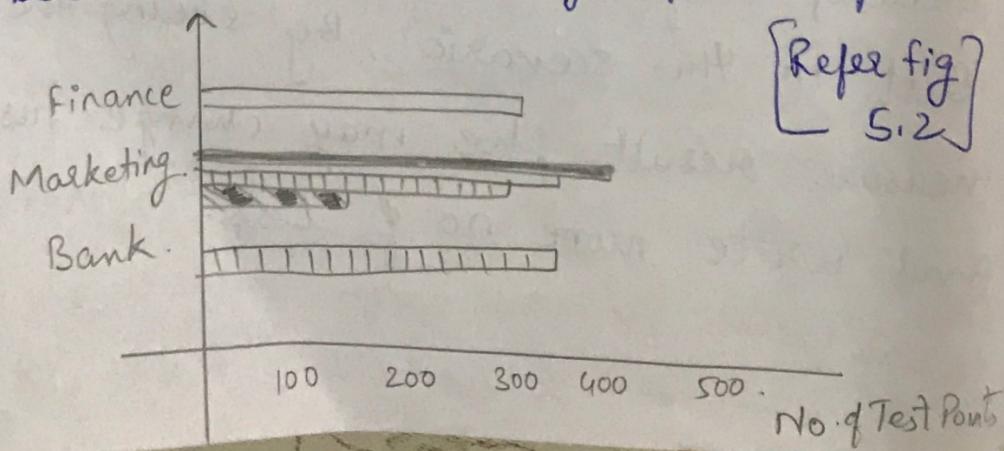
[Refer Fig 5.1]



- Successful Test Points
- — — Test steps that did not run
- - - Failed Test Points

- The no. of Test Points should be increased as the product is developed. (—■—)
- The failed Test Points should be decreased as the product is being developed (- - -)
- We can also drill down into the data to observe the product size per component.

[Refer fig 5.2]



② Pulse

- The aim is to measure the continuous Integration.
- The data required to measure this component is gathered from the Development environment by counting the no. of check-In operations occur per day.

- The Data is gathered for code check-Ins, automatic acceptance test check-in's, and detailed specifications check-in's.

- But the 2 risks with this measure:

^{risk}
① Some of the team members can simply click for check-In operation, just to increase the pulse count.

- To overcome this, check-In operations are counted only when there is a change in the integrated part and it should reflect the continuous integration.

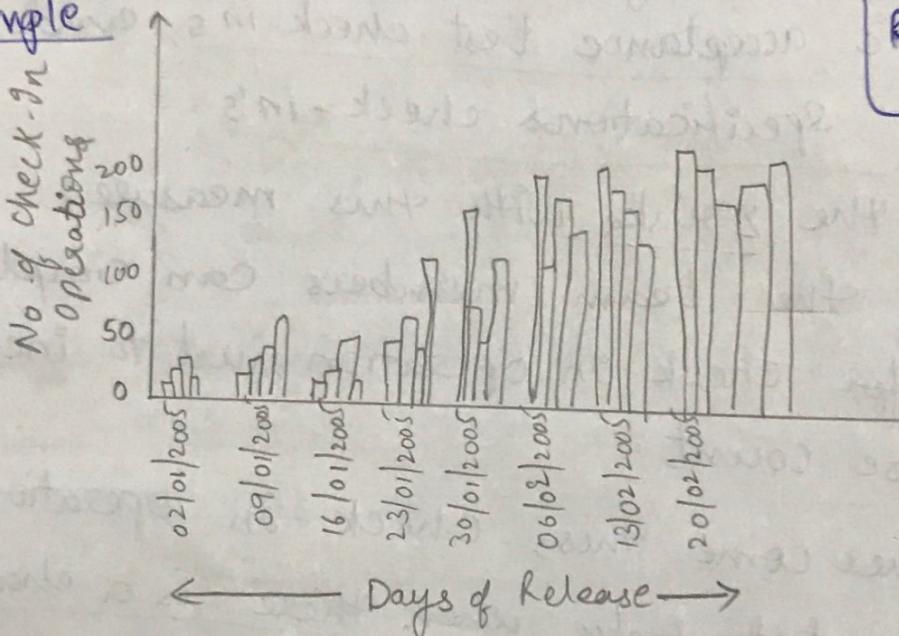
^{risk}
② Some of the team members may create overhead and this lacks continuous integration. A Team consists of people with different mindsets. They will not check-In the code daily.

- Daily Pulse will reduce the iteration overhead and bug fixing.

- There are 2 types of Pulse:

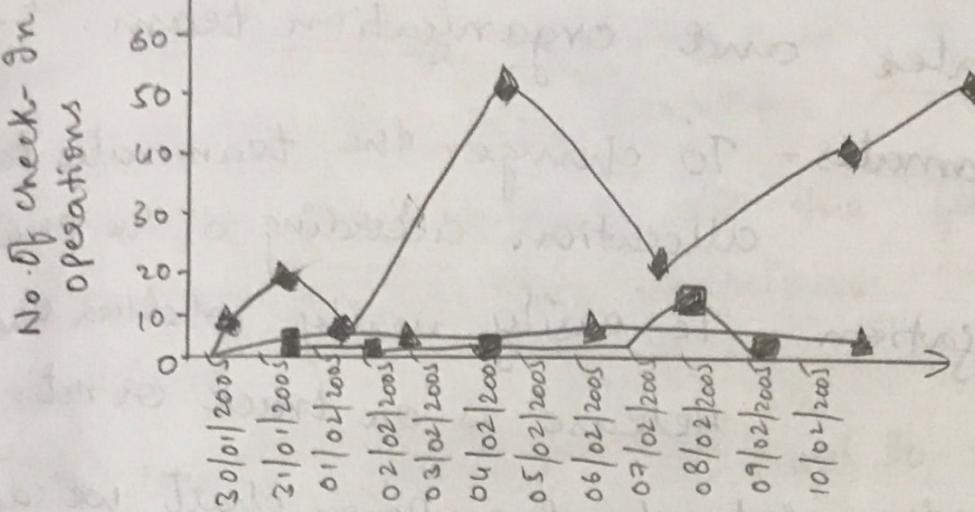
- STEADY: The pulse is more or less equal to the desired status.
- SPIKY: Most of the check-ins are grouped at the end of iterations, which means that these developers did not integrate the code during the iterations.
(This SPIKY reflects a negative sign).

Example



[Refer fig
S.3]

- Generally, first & second weeks of an iteration will have a fewer check-ins. During 4th iteration (7 & 8 weeks), the work has to be distributed among iteration days.
- We can also see the Pulse with respect to the type of check-ins:
 - (i) Code Check-ins (→)
 - (ii) Automated Test Check-ins (↔)
 - (iii) Detailed Specs Check-ins (↑)



③ Burn - Down

- The Burn- Down is a chart that presents the project's remaining work versus the remaining human resources.

- for this chart we need 2 Tables:

① Planning Table:

This table maintains data for each task like opening date, closing date, estimated development time , real development time, etc

② Human Resources Table:

This table maintains the data like allocation of work to each teammate, what is his/her portion percentage in the project

- The Burn-Down chart will be presented per week till the release is completed.

- This Burn Down chart is useful to both teammates and organization team.

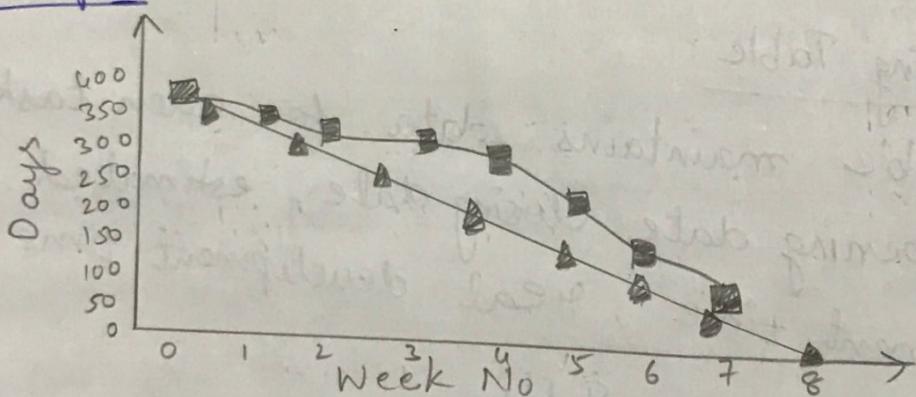
→ For Teammates - To change the teammates tasks allocation according to the priority.

→ Organization - To easily verify whether the release is on track or not.

- With the help of Burn Down chart we can know whether the project plan can actually be accomplished ~~or not~~.

- The Remaining work and Remaining resources should reduce for every week of development.

Example



— Remaining Work

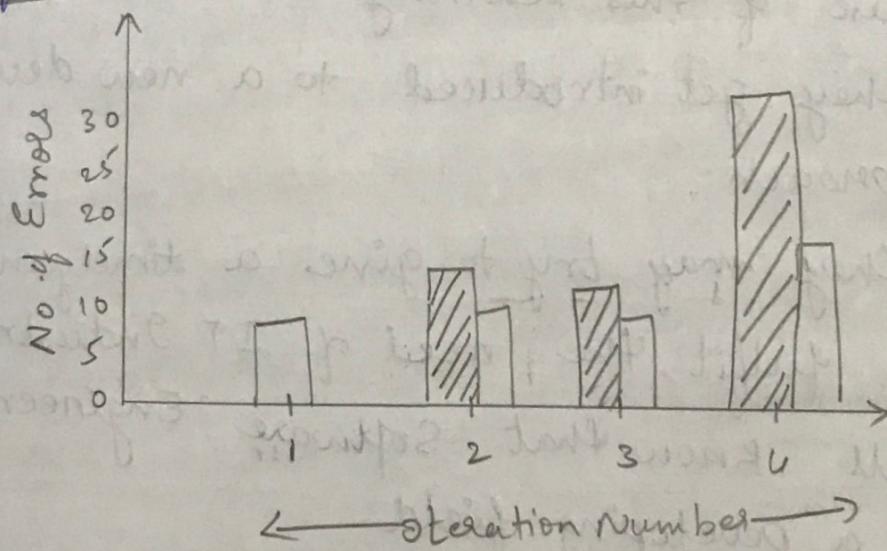
— Remaining Resources.

- For this project, initially the resources were 387 Days for entire release while work is estimated for 370 Days.

- As the week goes on, the remaining work and remaining resources get decreased.

④ Faults

- The aim of this measure is to count the no. of faults per iteration.
- During the new release, all the faults that were discovered in the previous iteration were fixed.
- This Faults measure is used to measure the product quality. [It is a standard Quality Measure]
- The faults can Coding errors or detailed specification errors.



Code Errors



Detailed Specs Errors

Please be careful for every graph about the x-axis & y-axis. Each measure will be having different attributes on its axis.

Measures in Learning Environments (3 ways)

① Teaching & learning Principle: Emphasize the slow Development Approach in the context of world of slow Development

- It is always preferred to connect the approach to the world of slow development

Ex: Take the problems faced by IT Industry. Explain them with example about the problem, let others learn about the problem and Try to overcome it.

- Because of this learning:

① They get introduced to a new development approach.

② They may try to give a timely answer to fulfil the need of IT Industry.

- We all know that Software Engineering is still a developing field.

② Measurement Activities

- There are 4 Activities

① Define the Measures

② Decide on how to collect the relevant Data

③ Analyze & Present the Measures

④ Assess the project status by using Measures

① Measure Definition

- Write down the goals that are important for your project success.
- Open discussion about the Project goals and post them on white board.
- Break Down the goals into measurable sub-goals
- Again have an open Discussion about the subgoals.
- Exchange the sub-goals among groups and get suggestions about measures
- Discuss about the measures posted on white boards with respect to title, description, relationship to main goals and sub-goals, ways to collect the data for the measurement & teammates who will be the in-charge of this measurement. (Overall Discussion)

② Data Collection

- Team Members who are responsible for the measures should explain to the other teammates about what data they expect to report.
- How to gather the information manually

or how to get the information automatically

③ Measure Analysis & Presentation

- The measure analysis and presentation will be carried by Tracker and the team members who are responsible for a specific measure.
- The presentations can be:
 - weekly (stand-up Meetings)
 - iteration wise (iteration planning)
 - release end (end of release)

④ Assessment

- Check whether the measures reflect to the project's goals & sub-goals.
- If measures is not reflect to the goals, then we need to refine | change the measures.

③ Role-Related Measures

- For Role related scheme, there are 3 main Measures:

- ① Role Time Measure
- ② Role Communication Measure
- ③ Role Management Measure

① Role Time Measure

- It measures the ratio of development role to role performance. i.e. the time invested in the development task relative to the time invested in role activities.

Ex:

Explain
this table
in words
also

Person	Time spent on Development (%)	Time spent on Role (%)
A	70%	30%
B	60%	40%
C	80%	20%

② Role Communication

- It measures the level of communication in the team at each development stage.

- Each role-holder needs to communicate with other team members in order to perform his/her tasks efficiently.

Ex: Role communication can be measured by examining the electronic forum.

- 100 messages were sent during 1st iteration.
- Explain what in Table words also*
- 80 were sent by the developers.

Role	Forum Messages Total	Forum Messages (%)
Leading Group		
(a) Coach	225	32.2
(b) Tracker	118	17
Customer Group		
(a) Customer	32	4.5
(b) Acceptance Tester	14	2
Maintenance Group		
(a) Presenter	53	7.6
(b) Documenter	53	7.6

③ Role Management

- It measures the level of Project Management
- The maximum level is obtained when all role-holders provide maximum role performance. (Participation)
- This measure can be calculated by summing the total no. of messages during all iteration weeks and examining them (Both the original messages posted & their responses.)