ILLINOIS TECH | College of Computing

ITMD 536 Software Testing & Maintenance

Chapter 11 and 12

Resourcing & Measurement Data Utilization

Objectives

- What are the best practices for estimating and budgeting?
- What are necessary skills, knowledge and abilities?
- What is facility optimization and utilization?
- How can you do focusing on workload load balancing?
- What is data, when and why do you need?
- How can you use the quality insights using defect data?
- How can you use productivity insight using cost data?
- How can you use management insights using process feedback?

Objectives

- What is configuration & version management?
- What is process, service, and software quality assurance?
- What is maintenance measurement and analysis?
- What is casual analysis & problem resolution?
- What is software rejuvenation, migration, and retirement?

11 Proper Resourcing (Staff and Equipment)

- **11.1 Estimating/Budgeting Best Practices**
- The process of estimating software development costs starts with requirements.
- An average software productivity rate of one SLOC/staff-hour, and a cost of $50/staff-hour.

11.1 Estimating/Budgeting Best Practices

- The cost/staff-hour might have to be raised to accommodate a fast-paced schedule where people have to work overtime to get the job done and the size estimate may vary plus or minus by 30 percent.
- There are several methods and models to predict project costs and duration using requirements as their basis.
- Popular models used to do estimates are COCOMO II, SEER, SLIM, True S (PRICE).

11.1 Estimating/Budgeting Best Practices

- The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics.

11.1 Estimating/Budgeting Best Practices

- SEER for IT enables organizations to develop an early, accurate assessment of costs, schedules and risks for IT projects and their operations, helping to maximize productivity and output with fixed or declining budgets. SEER for IT has a collection of IT estimation elements (process models) which can be used.

11.1 Estimating/Budgeting Best Practices

- **SLIM-Estimate**® helps you estimate the cost, time, and effort required to satisfy a given set of system requirements and determine the best strategy for designing and implementing your software or systems project.
- In addition to software cost estimation, this powerful systems and software project estimation tool provides a **high level of configurability** to accommodate the different design processes being used by developers today: such as

11.1 Estimating/Budgeting Best Practices

- Agile development, business intelligence, package implementation, hardware, call center development, infrastructure, model-based development, engineering and architecture design, service-oriented architecture, SAP, Oracle, and more.
- PRICE® Cost Models™—based on historic project data verified and validated by extensive PRICE® Research™ analysis and refinement—enable estimators to generate cost proposals using reliable top down estimating based on correlations to the scope and complexity of new projects.

11.1 Estimating/Budgeting Best Practices

- This model helps generate estimates for cost proposals in hours instead of weeks or months. It saves time and money in resource planning management, and provide a sound basis for program maintenance activities.
- This also project future costs of currently immature technologies, based on engineering complexity, engineering skills, and scope of design efforts.

11.1 Estimating/Budgeting Best Practices

- **Poor Person's Guide to Estimating Development and Maintenance Costs:** (13)
- **1. Define the work elements:**
  - (Development)(D): Main focus is on development, but management and support tasks can take a significant part of the effort
  - (Maintenance) (M): Focus is on both maintenance and sustaining engineering with both taking about equal parts of the effort to perform for the "to be fielded" and "fielded" releases. New releases are distributed about 70/30. (maintenance sustaining)

11.1 Estimating/Budgeting Best Practices

- **2. Determine the software size:**
  - (D): Size software product from specification in either source lines of code (SLOC) or function points; address platform updates and legacy and commercial off-the self (COTS) as part of the process.
  - (M): Determine the number of software change requests (SCRs) that need to be processed by looking at logs and backlog; address platform updates and legacy and COTS

11.1 Estimating/Budgeting Best Practices

- **3. Assess the program difficulty:**
  - (D): Specify one of three levels of product difficulty to be used to make adjustments (easy, moderate, and hard)
  - (M): Specify one of the three levels of product difficulty to be used to make adjustments (easy, moderate, and hard)

11.1 Estimating/Budgeting Best Practices

- **4. Adjust for personnel capabilities**
  - (D): Again, specify one of three levels of staff skills and experience to be used to make adjustments (low, nominal, and high mix)
  - (M): Again, specify one of three levels of staff skills and experience to be used to make adjustments (low, nominal, and high mix)

11.1 Estimating/Budgeting Best Practices

- **5. Make other adjustments:**
  - (D): Adjust other factors that cause the cost to vary more than the norm; process maturity is the example explained in the Guide
  - (M): Adjust other factors that cause variation in cost; availability of facilities and tools has a great influence on maintenance costs as does testing

11.1 Estimating/Budgeting Best Practices

- **6. Estimate effort:**
  - (D): Estimate effort by multiplying a base productivity number (included in the Guide) times the size, taking adjustments into account during the calculation
  - (M): Estimating the number of SCRs that can be processed for your fixed budget using productivity numbers available (number of hours/productivity rate)

11.1 Estimating/Budgeting Best Practices

- **7. Adjust for risk:**
  - (D): Develop a range of costs for your estimate that take factors like requirements volatility and size growth into account
  - (M): Develop a range of costs for your estimate based on factors that influence your ability to get additional SCRs processed
- **8. Estimate duration:**
  - (D): Estimate duration using formula provided in the Guide
  - (M): Schedule fixed and cannot vary because it is secured in cement

11.1 Estimating/Budgeting Best Practices

- **9. Validate reasonableness:**
  - (D) Use cost and productivity benchmarks (some provided in Guide, using your own past performance recommended) to verify reasonableness of your cost and productivity numbers
  - (M): Use past performance as your primary benchmarks for verifying the reasonableness of your cost and productivity numbers; if external numbers are available, use them as well.

11.1 Estimating/Budgeting Best Practices

- **10. Extrapolate the estimate:**
  - (D): Extrapolate the estimate to determine what it will take to accomplish other tasks that need to be done (support tasks like quality assurance [QA], system test, etc.) using percentages provided in the Guide for that purpose; these tasks can cause your estimate to double in certain cases
  - (M): Extrapolate the estimate to determine what it will take to accomplish other tasks that need to be done (independent testing, sustaining, engineering, etc.) using percentages developed for that purpose; at the least, double your estimate to account for sustaining engineering tasks for all releases that need to be supported

11.1 Estimating/Budgeting Best Practices

- **11. Staff-load the schedule:**
  - (D): Estimate the size of your team (divide total hours/months) and allocate staff to schedule by task using a spreadsheet as your tool; adjust staff loading so that it evens out across the schedule
  - (M): Team size is for the most part fixed in maintenance; allocate staff to scheduled by task using a spreadsheet as your tool; adjust staff loading so that it evens out across the schedule

11.1 Estimating/Budgeting Best Practices

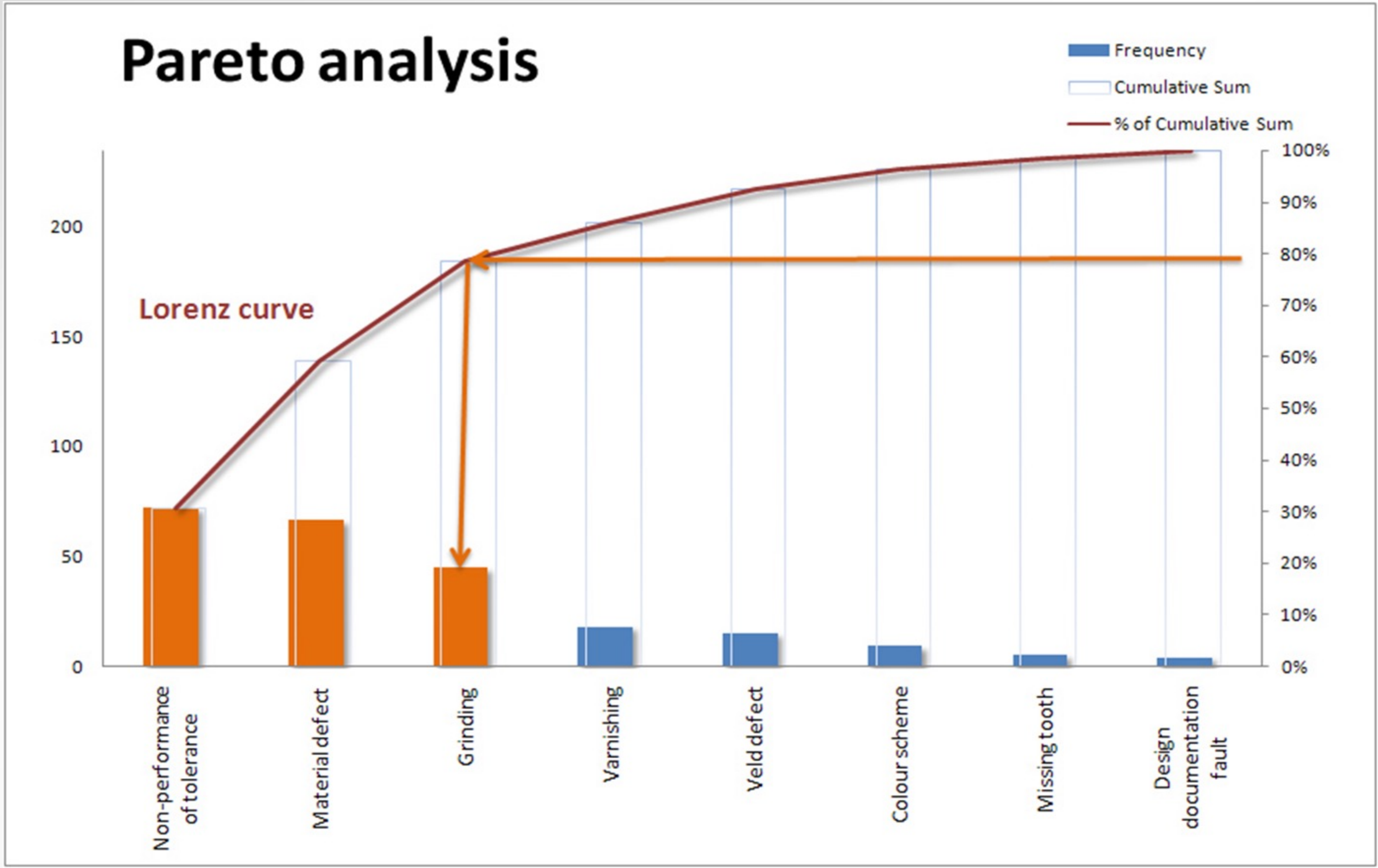
- **12. Reiterate until satisfied:**
  - (D): Compare results against expectations and reiterate by making adjustments to factors such as size and difficulty as you go through the process again until an acceptable cost/schedule solution can be reached
  - (M): Compare results against expectations and reiterate by making adjustments to your staff allocations until an acceptable solution can be reached; use the load balancing model presented in this chapter to help

11.1 Estimating/Budgeting Best Practices

- **13. Update periodically:**
  - (D): Keep estimates up to date by presenting cost and schedule-to-completes at major reviews
  - (M): Keep estimates up to date by presenting cost-and schedule-to-completes at major reviews

11.2 Necessary Skills, Knowledge, and Abilities

- For maintenance, proven tools and techniques can be used to determine “difficult” and “error-prone” modules.
- Other metrics should be considered approaches such as defect data error-prone modules like Pareto analysis.


Pareto analysis
Frequency
Cumulative Sum
% of Cumulative Sum
Lorenz curve
0
50
100
150
200
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Non-performance of tolerance
Material defect
Grinding
Varnishing
Veld defect
Colour scheme
Missing tooth
Design documentation fault

11.2 Necessary Skills, Knowledge, and Abilities

- Use small and well-organized teams that work with a minimum of interference
- Staff the project with personnel who can communicate well and work together well
- Staff the project with personnel who has high degree of system knowledge
- Staff the project with personnel with knowledge of the platform and operating system
- Staff the project with personnel with knowledge of the programming languages, tools and tool environment and who know how to use them to get results

11.3 Facility Optimization and Utilization

- Facilities for development are often dedicated, but those for maintenance many timeshare time and space with operational personnel who use them for training exercises, and host of other tasks.
- Determine the optimal mix of work for the facilities requires a sophisticated mathematical model that can handle the many different states a facility can be in at a given point in time and different types of cots that the holder may incur depending on the state.
- https://www.youtube.com/watch?v=gY1RNbOVRrU

11 Proper Resourcing (Staff and Equipment) Allocation of Facility Time by Percentage of Work Activity

Activity	Usage Range
Test and Integration	55 to 70%
Emergency Repairs	6 to 10%
Prototyping and Special Studies	4 to 8%
Commercial off-the-shelf (COSTS) evolution	5 to 10%
Training	6 to 12%
Exercises and demonstrations	4 to 8%
Minor operations	6 to 12%
Preventive maintenance and repairs	4 to 8%
Total	90 to 138%

11.4 Focusing on Workload Load Balancing

- Maintainers get allocated fixed levels of staff that they had been budgeted by balancing the workload for their maintenance tasks.
- https://www.youtube.com/watch?v=CxapGqlh3Fg
- Projects were typically funded on a level-of-effort (LOE) basis.
- Project performed a range of tasks in addition to what most in the industry believed were the normal software maintenance activities. (sustaining engineers, acquisition management support, operations and testing (V/V)

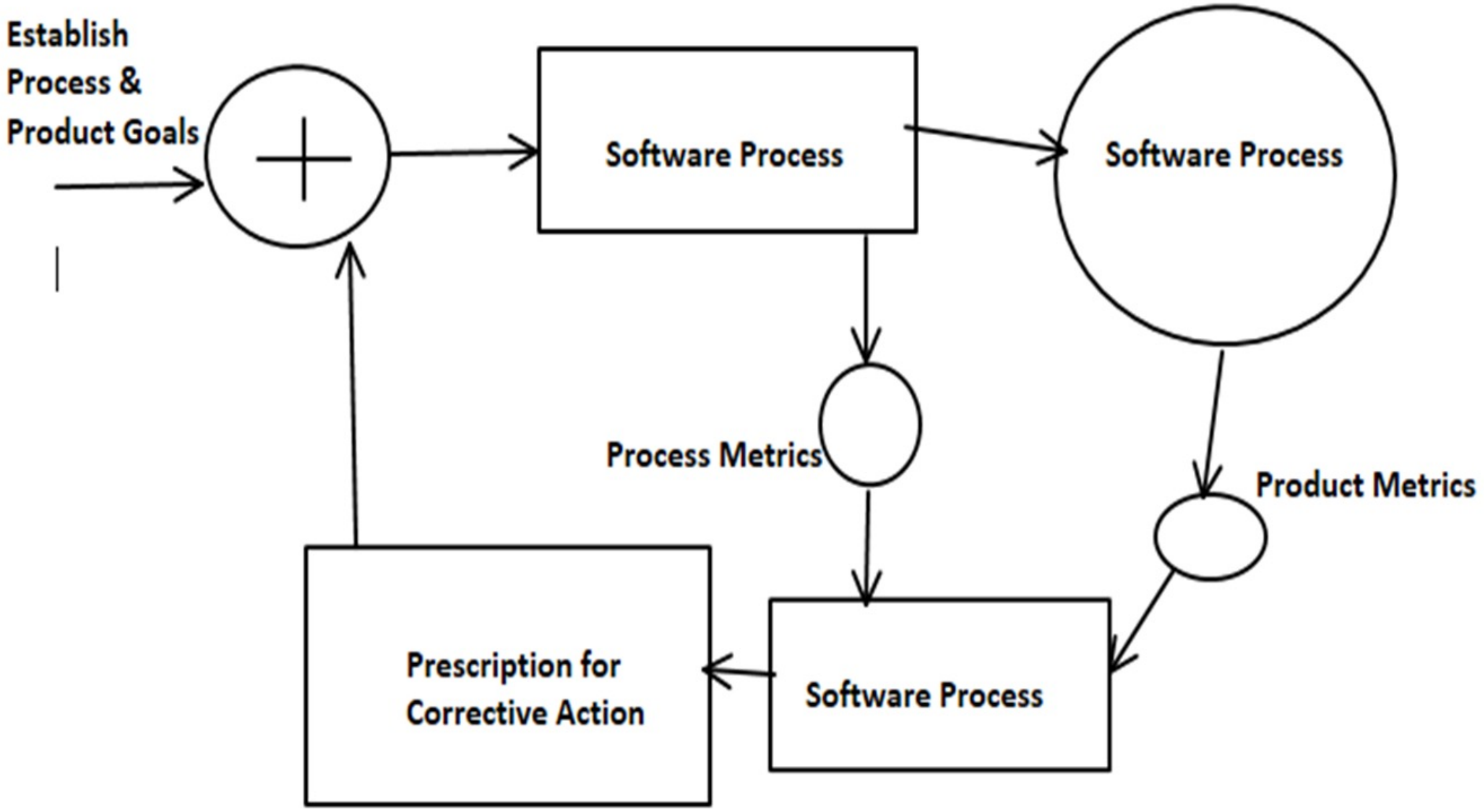
11.4 Focusing on Workload Load Balancing

- Facilities used to generate and test releases as well as support staffs are often maintained as part of the sustaining engineering task.
- The same situation existed for other tasks like independent testing, information operations, interoperability testing and acquisition support.

12 Effective Measurement Data Utilization

- **12.1 What Data, When and Why**
- The mature organization CMMI Level 3 will capture and use their measurement data at the enterprise level to assess their cost, productivity, and quality using six sigma and statistical process control techniques to determine whether or not they are achieving their business goals.

Measurement-driven management process for mature groups



12.1 What Data, When and Why

Eight areas of focus where metrics are measured:

- **1. Change Management:** refers to whether the software maintenance group has the ability to support desired changes to the software to alter capabilities or system capacities
- **2. Problem Resolution:** refers to whether or not the software maintenance group has the ability to support both repairs and changes to the system that properly address interruptions or reductions in service stemming from problems identified during operations

12.1 What Data, When and Why

- **3. Request Fulfillment:** refers to whether the software maintenance group has the ability to adequately handle requests for action made of the system by users during the course of operations (planned updates, moves of equipment, etc.)
- **4. Test Effectiveness:** refers to whether or not the test program established by the software maintenance group is effective in terms of its coverage and degree of automation; regression testing is high-lighted as a target of opportunity for improvement in most software maintenance groups because of its potential large impact on operations

12.1 What Data, When and Why

- **5. Customer Satisfaction:** refers to how satisfied the customer is with the overall service provided by the software maintenance group
- **6. Workforce Loading:** refers to whether the software maintenance group has the workforce capacity to adequately handle the desired workload (assignments by management)

12.1 What Data, When and Why

- **7. Equipment Loading:** refers to whether the software maintenance group has the equipment and facility capacity to handle the workload placed on its maintenance integration and test, and operational facilities
- **8. Help or Service Desk Efficiency:** refers to how well the help or service desk handles customer issues and requests of service

12.2 Quality Insights Using Defect Data

- The defect data that you are collecting can provide maintenance managers with meaningful insights in both product and process quality.
- Defect occurrences or number of defects by priority, age, and sometimes type (e.g., design or logic defect)
- Defect densities or number of defects per unit size measure (defects/function point, defects/KSLOC, etc.) by component by priority and age

12.2 Quality Insights Using Defect Data

- Defect rate or number of defects occurring per unit time by component by priority
- Defect data can be used effectively to assess the quality of the processes that you use the products that you generate.
- Defects measurements taken over a specified time period tell you whether or not the process is operating within control limits.
- Based on the results, you should investigate and resolve the sources of abnormality.

12.3 Productivity Insights Using Cost Data

- The cost data that you are collecting can provide meaningful insights into project performance and personnel productivity. Performance is judged in terms of delivery of an acceptable product on schedule and within budget.
- Productivity is measured in terms of output generated divided by the inputs needed to produce them (software change requests [SCRs/staff-hour, function points/staff hour, etc.)

12.3 Productivity Insights Using Cost Data

- When managing a software project (maintenance or development), there are things that you can control and other s that you cannot like facilities.
- Look for parameters that influence cost and productivity that you can control are:
  - Product parameters such as complexity, platform volatility, and amount of documentation

12.3 Productivity Insights Using Cost Data

- Personnel parameters such as workforce capabilities, degree of teamwork, turnover rates, and experience
- Process parameters such as maturity level
- Project parameters such as degree of tooling, degree of colocation, and amount of schedule compression
- Keep it simple

12.4 Management Insights Using Process Feedback

- Institutionalizing organizational processes for measurement is part of both the International Standards Organization (ISO) and CMMI frameworks.
- The CMMI has two process areas that directly links to measurement in a mature environment: quantitative project management (QPM) and measurement and analysis (MA).
- The goal of QPM process area is to quantitatively mange the project's defined process in a manner that enables it to achieve its quality and process performance goals.

12.4 Management Insights Using Process Feedback

- The MA process area abets and assists by developing and sustaining the measurement capability that gathers the information needed to determine whether management goals have been fulfilled.
- https://www.youtube.com/watch?v=__IlYNMdV9E
- https://www.youtube.com/watch?v=SBlKdEFAnlM

12 Effective Measurement Data Utilization–Goal–Improve Quality

1. How to define quality
2. Who are the users – how to present data
3. How do I measure quality for each of these classes of users?
4. What data do I need to collect to quantify these measure

1. Customer satisfaction
2. Senior managers – Monthly reports
   Project leads – incentive plans
3. Senior managers – complaint rates
   1. Project leads – Defect rates
   2. Defect densities
   3. Defect backlog
4. No of complaints, response time, resolution time etc.