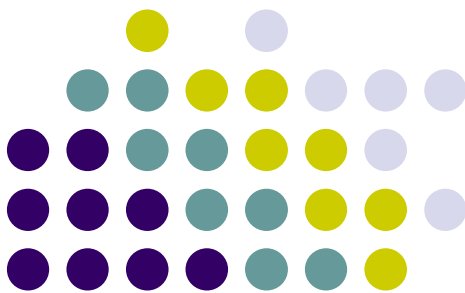


Metrics

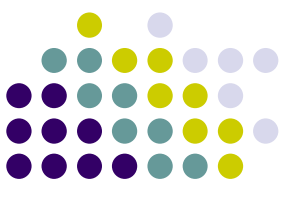




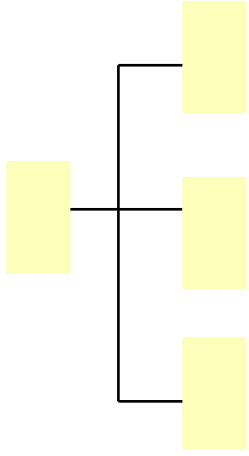
Why Measure

- **Basic Project Control**
 - Planning Effort and Timescales
 - Monitoring Progress
- **Quality Assurance**
 - Identify Problems
 - Confirming Achievement of Requirements
- **Process Improvement**
 - Identify Process Problems
 - Evaluate Process Change Options
 - Monitoring Process Change

Which Attributes/ Metrics?



- Depends on Business and Product Goals
- Some Common Management Goals
 - Keep to Budget/Timescales
 - Minimise pre and post release defects

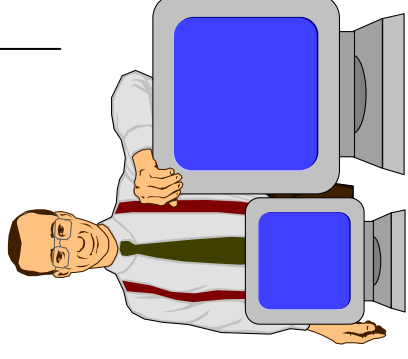
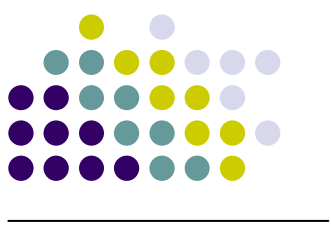


Quality Model

Quality Characteristics

Quality Attributes

External measures

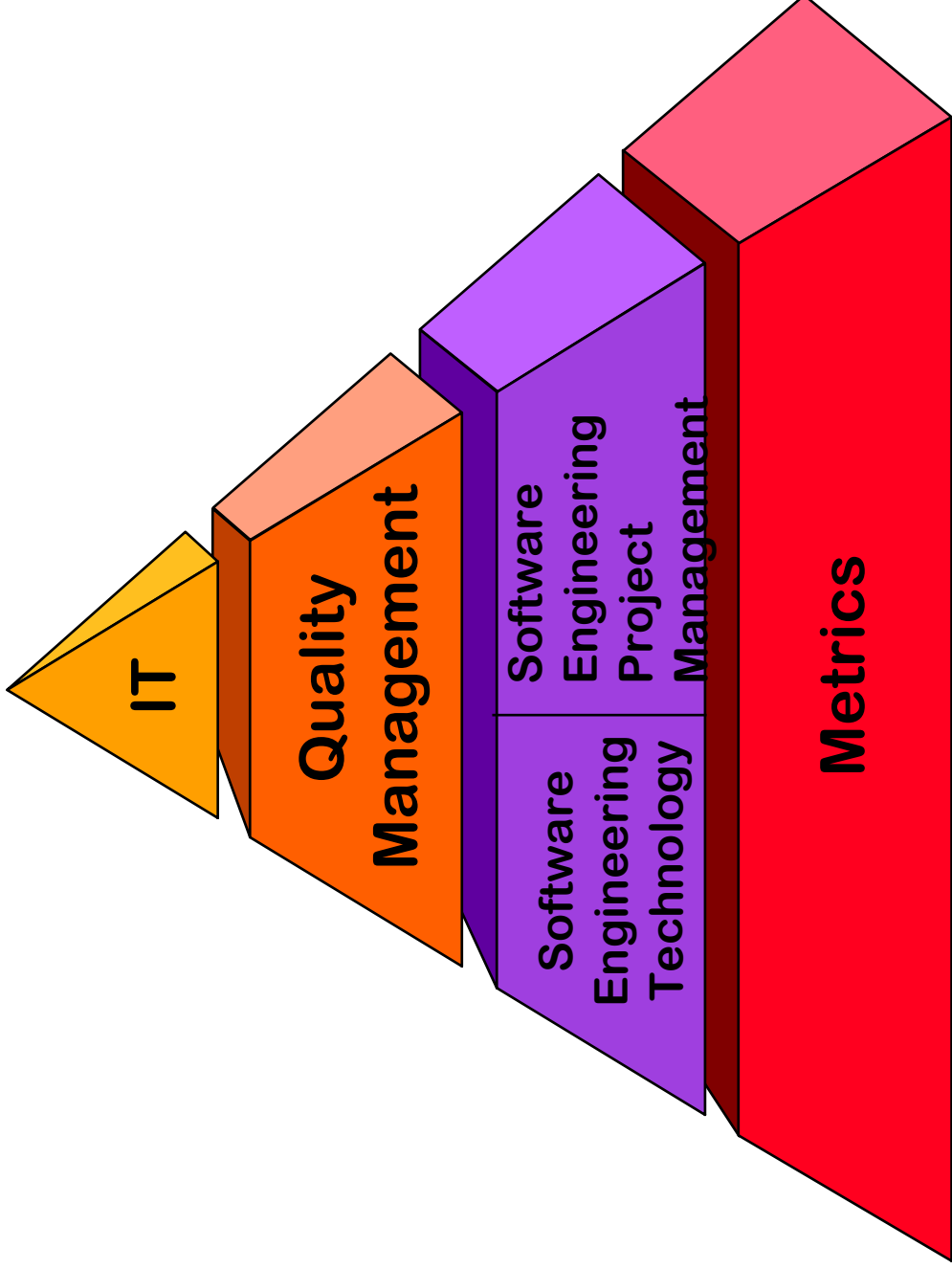


Development Process

Internal Measures

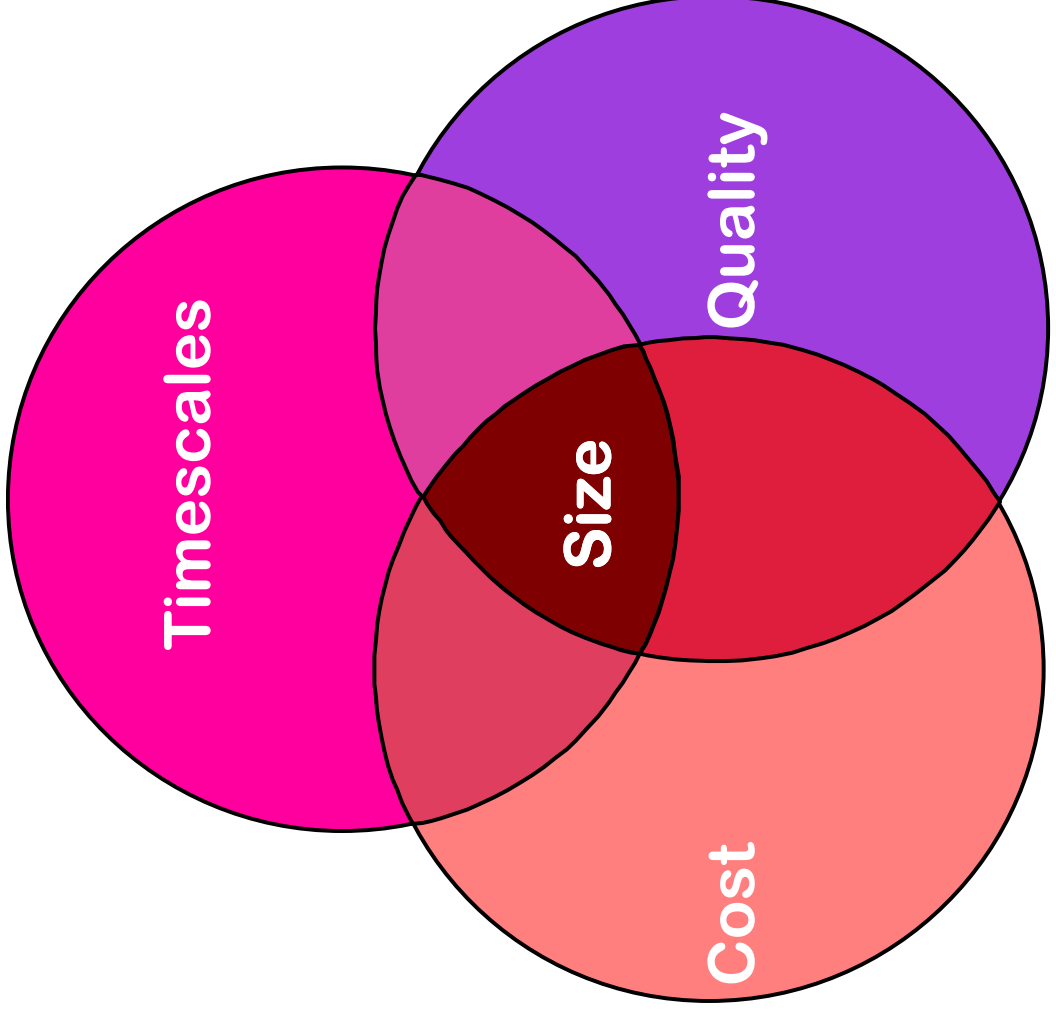
Controls on process to deliver product.

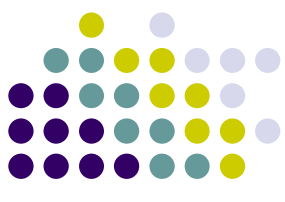
The Foundation





Balanced Measures

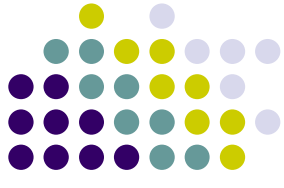




External Measures

- To improve productivity
- To estimate costs more accurately
- To measure customer satisfaction
- To know how effective my department is in comparison to the rest of industry

Internal Measures



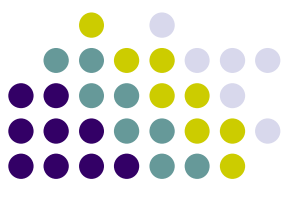
- To control the engineering of the product by measuring the process and partial products
- To obtain early warning of possible potential breaches of the project envelope



Problems with Software

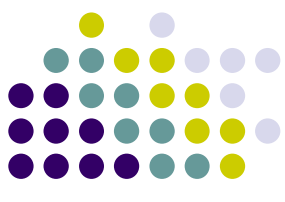
- Many different causes for deviation
 - Not always a problem
 - Small effort expenditure
 - Easy Task?
 - Poor Quality Work?
 - Planned Value can be wrong
 - Many responses to each cause

Planning for Quantitative Control

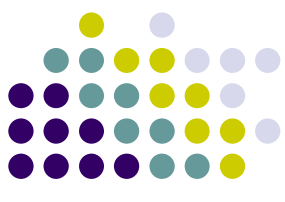


- Establishing attribute values for monitoring
- External Requirements
 - Budget, Delivery Date, Quality
 - Change only by negotiation with client
 - Data Analysis used to assess feasibility

Planning for Quantitative Control

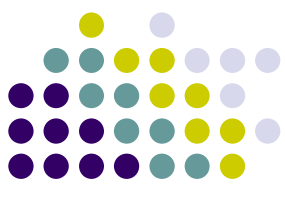


- Milestone Plans
 - Effort, date achieved, defects, changes
 - Expected Value
 - Upper and Lower Bounds
 - Derived from requirement and previous experience



Monitoring Project Progress

- Project Model is a Prerequisite
 - Phases and Activities
 - Milestone related products
- Phase/Activity relationship
 - Many Activities per phase
 - Activities span phases
- Quantitative Project Monitoring
 - Easiest when many projects/subprojects use the same model



Measurement Programme

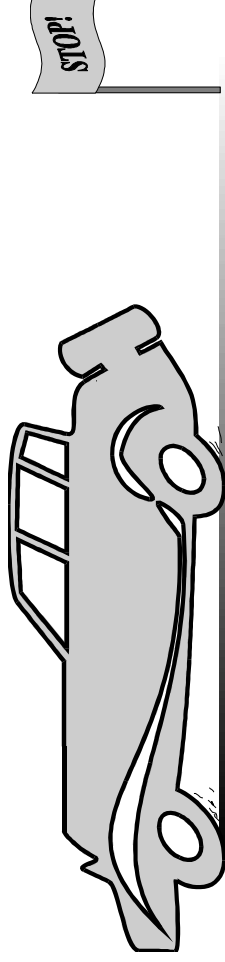
- Analyse what you need to know
- Define the data
- Decide
 - Where
 - Who
 - How
- Collect the data
- Analyse and interpret the data

Measurement



Example:

“Breaking distance of a car”



Measurement

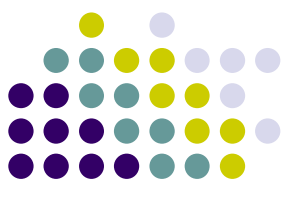
“Breaking distance of a car”

Attribute: Breaking distance

Evaluation Object: Car



Measurement



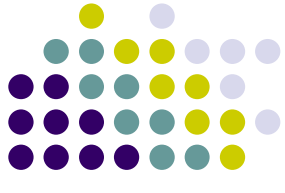
Measurement definitions:

Attribute: A measurable property of an object.

Evaluation Object: The object that possess the property to be measured.

Unit: The way to quantify the measurement.

Counting Rule: Conditions and procedures under which the measurement is obtained.



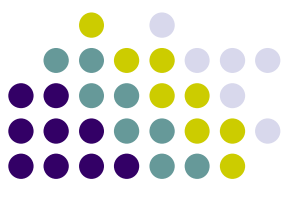
Measurement

Measurement Guidelines:

Attributes should be defined independently of Evaluation Objects, Units, and Counting Rules.

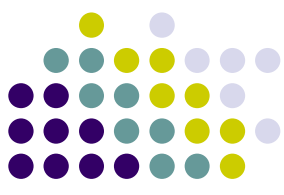
Units should be basic (e.g. hours, modules, faults) to allow for calculations. They have one of the following scale types: Nominal, Ordinal, Interval, or Ratio.

Counting Rules shall define measurement conditions (i.e. other attributes whose value influence the measurement value) and the measurement procedure in order to ensure that measurement values are repeatable and comparable.



Measurement

Attribute	Evaluation Object	Unit
Size	Project Requirements Spec. Requirements Spec. Detailed Design Doc. Module Module Development Team	'Small', 'Medium', 'Large' Requirements Function Points Modules Lines of Code Statements SW. Developers



Problem Report

Problem Report no.: _____ Date: _____

Project: _____ Sub Project: _____

Description of problem:

Development activity that detected problem: _____

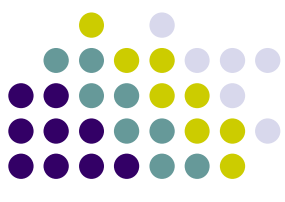
Fault severity: Major__ Minor__ Negligible__

Estimate effort to correct (hours): _____

Fault type: Req. __ Specs. __ Design __ Code __ Doc. __ Other __

Modules changed:

Effort to correct (hours): _____ Date: _____



Problem Report

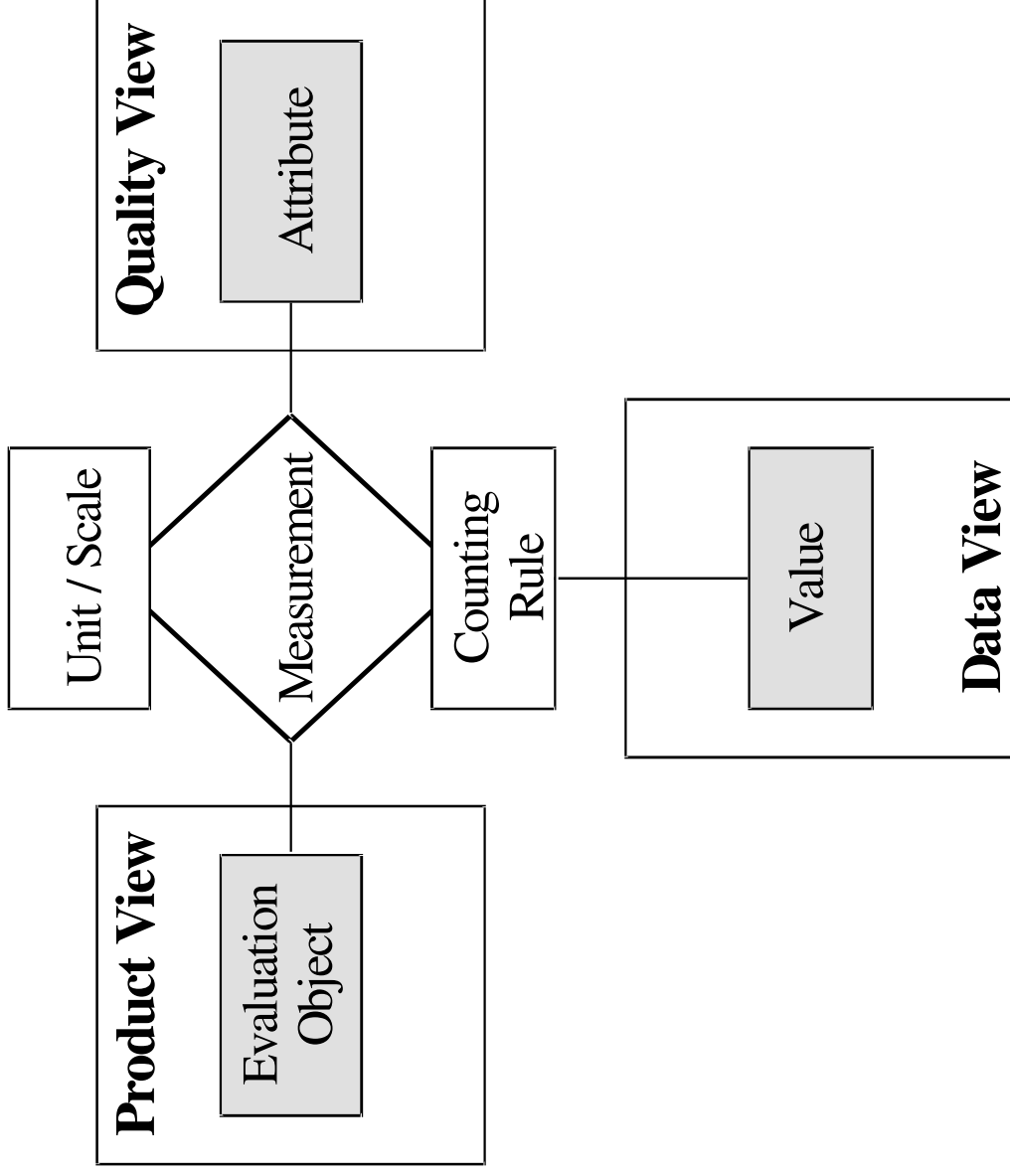
Attribute	Evaluation Object	Unit
Fault severity	Problem Report	'Major fault', 'Minor fault', ...
Fault type	Problem Report	'Req. fault', 'Spec. fault', ...
Correction effort	Problem Report	Manhours
Fault location	Problem Report	Modules

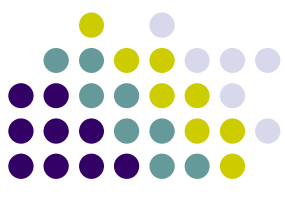
Problem Report



Link Type	From	To
Handled_by	Problem Report	Requirements Definition Functional Spec. Design Coding Integration Testing
Uses	Problem Report	Module
Groups_deliver.	Delivery of Reqr. Delivery of Func. Spec. Delivery of Sys. Design ...	Problem Report

Collation





Data Types

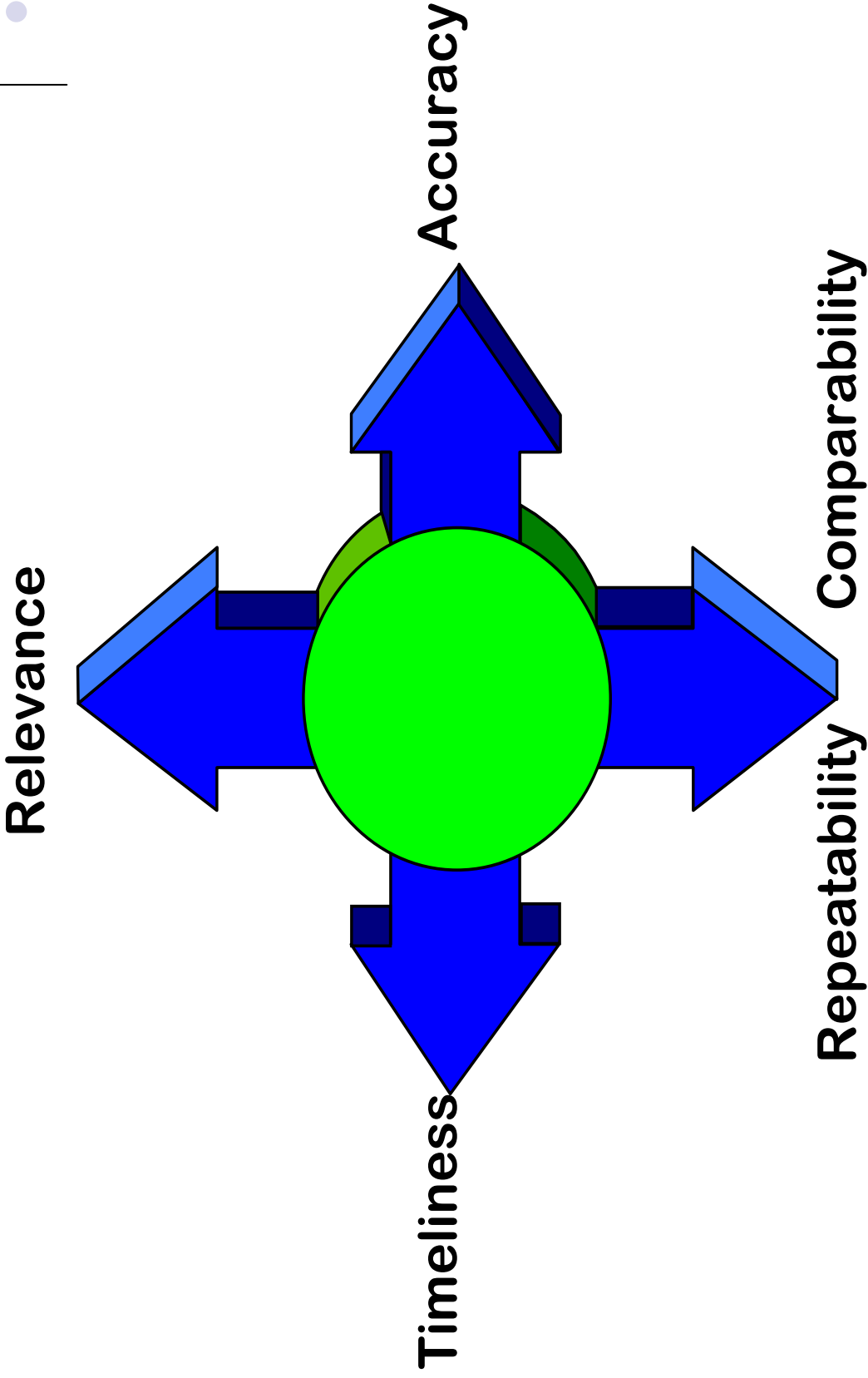
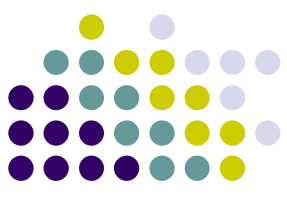
- **Ordinal Scales**
An ordered set of categories
- **Nominal Scales**
A non-ordered set of classifications
- **Ratio Values**
Usual counts and percentages
- **Date**
- **Character String**
- **Free Format Text**

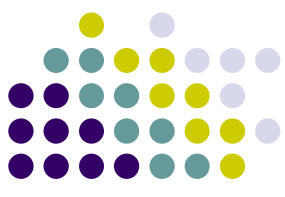


An additional dimension

- **Targets**
Edicts from above, these don't usually change
- **Estimates**
Usually provided by project, these will change through the life of a project
- **Actuals**
The actual value that occurred, these should not normally change

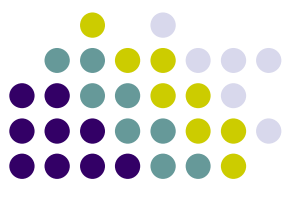
Useful Data





Useful Data Collection

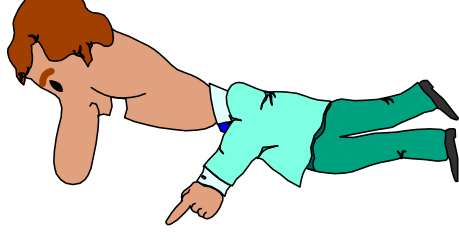
- Define information needs
- Define required measurements
- Integrating collection systems
- Motivating staff
- Getting started



Rule Number 1

- If you do not know why you are collecting a data item, or what you are going to do with the results:

**Do not
collect it**

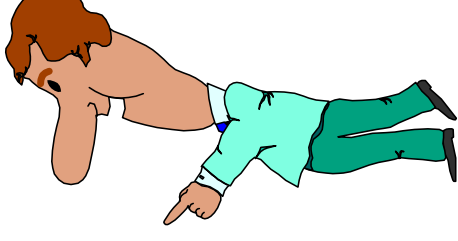


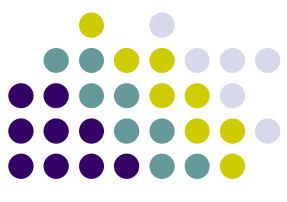
Rule Number 2



- The literature is full of far out metrics, if you don't understand it then you can't use it

**Use simple
measures**

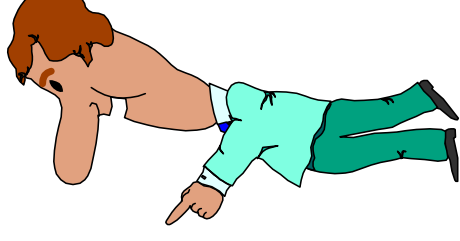




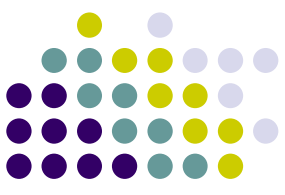
Rule Number 3

- Do make sure you know what your goals are, otherwise it is difficult to know if your measuring the right attributes

**Know your
goals**

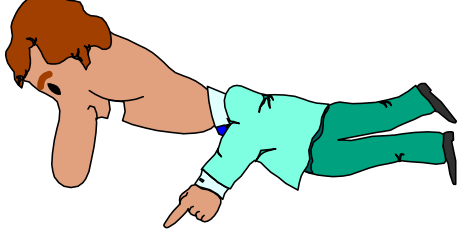


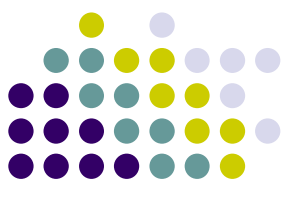
Rule Number 4



- Don't forget to verify that your goals are complete. You must measure all dimensions: cost, time and quality. If you attempt to solve quality problems and ignore timescales, guess what happens

**Cover all
dimensions**

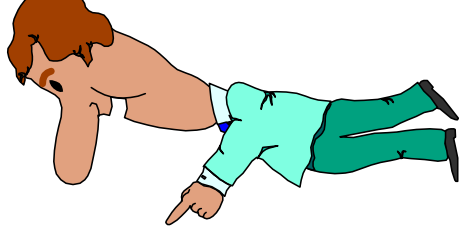




Rule Number 5

- Don't place too much reliance in industry norms. Results are only comparable if they use the same counting rules as you to the same constraints

**Trust your
data**

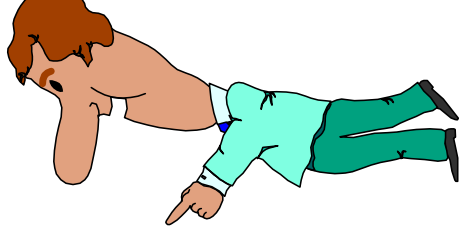




Rule Number 6

- Don't ignore the problem of motivating staff, if they don't know why data is being collected they will assume the worst

**Motivate
staff**

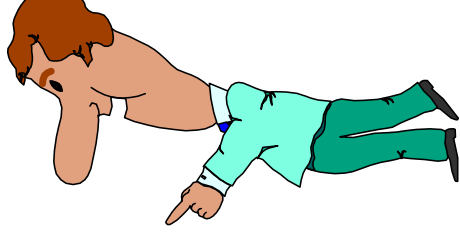


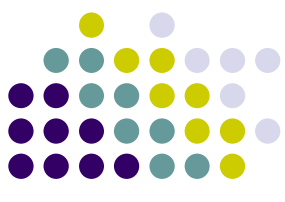


Rule Number 7

- Do plan your data collection around your development processes and your existing tools

**Fit to your
process**

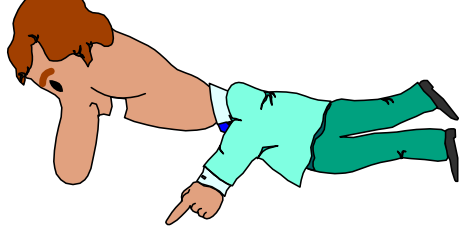


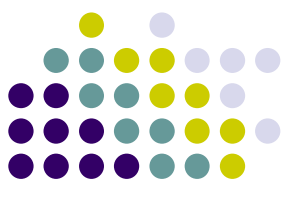


Rule Number 8

- Do take a realistic view of the costs of data collection. If it is fitted around the process it doesn't cost too much. It does cost to provide storage, data analysis and extraction capabilities and to train staff

**Be realistic
about costs**

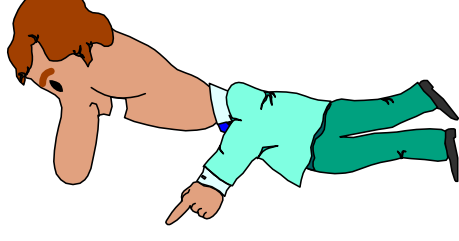




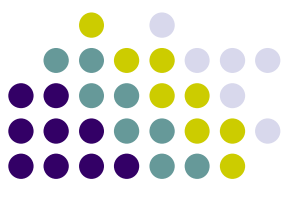
Rule Number 9

- Don't expect the metrics to solve all your problems. If your process is chaotic, your staff are demotivated, your requirements are unrealistic then metrics are nearly damage limitation tools

**Be realistic
about metrics**

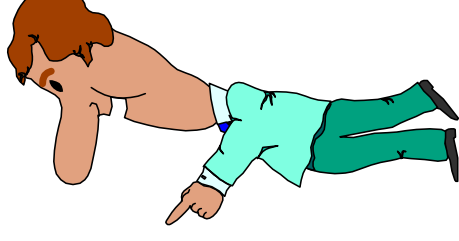


Rule Number 10

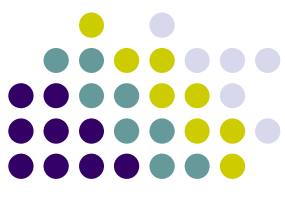


- Don't put off getting started. Every year for the last ten years someone says its too late we won't see any benefits for a year. If you had started

**Get started
now**



Which Attributes/ Metrics?



- Depends on Business and Product Goals
- Some Common Management Goals
 - Keep to Budget/Timescales
 - Minimise pre and post release defects



Phase-based size and structure metrics

An Ideas

