

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenter individually and are not the opinion or position of ICON. ICON does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

What can we learn from the lean six sigma process?

Have we used this methodology to improve the programming process?



Lean Six Sigma



Core objectives of Lean Six Sigma



Customer satisfaction

Delight them. Don't just hit targets, exceed them.



Adding value

Everything you do adds value in the eyes of the customer.



Eliminating waste and improving flow

If an activity doesn't add value or satisfy customers – remove it.



Capitalising on opportunities

Profit from overlooked value-add activities.

Complementary goals of “Lean” and “Six Sigma”

LEAN

A set of principles and tools that help us to eliminate process activities that don't add value (waste) and to improve “flow”

Efficiency

SIX SIGMA

A collection of methods and tools aimed at the significant reduction of unwanted variation in products and processes to improve yield

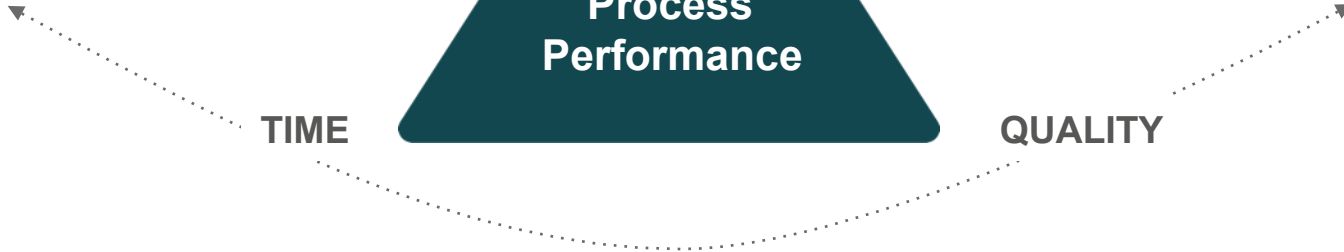
Effectiveness

COST

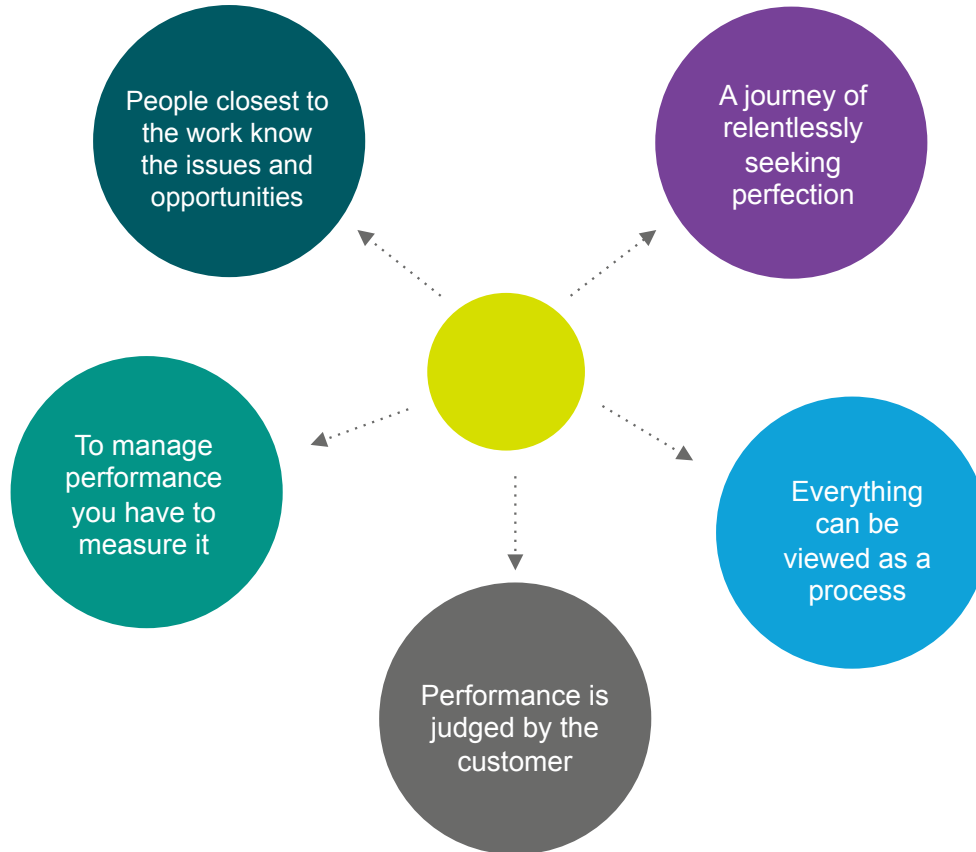
**Improved
Process
Performance**

TIME

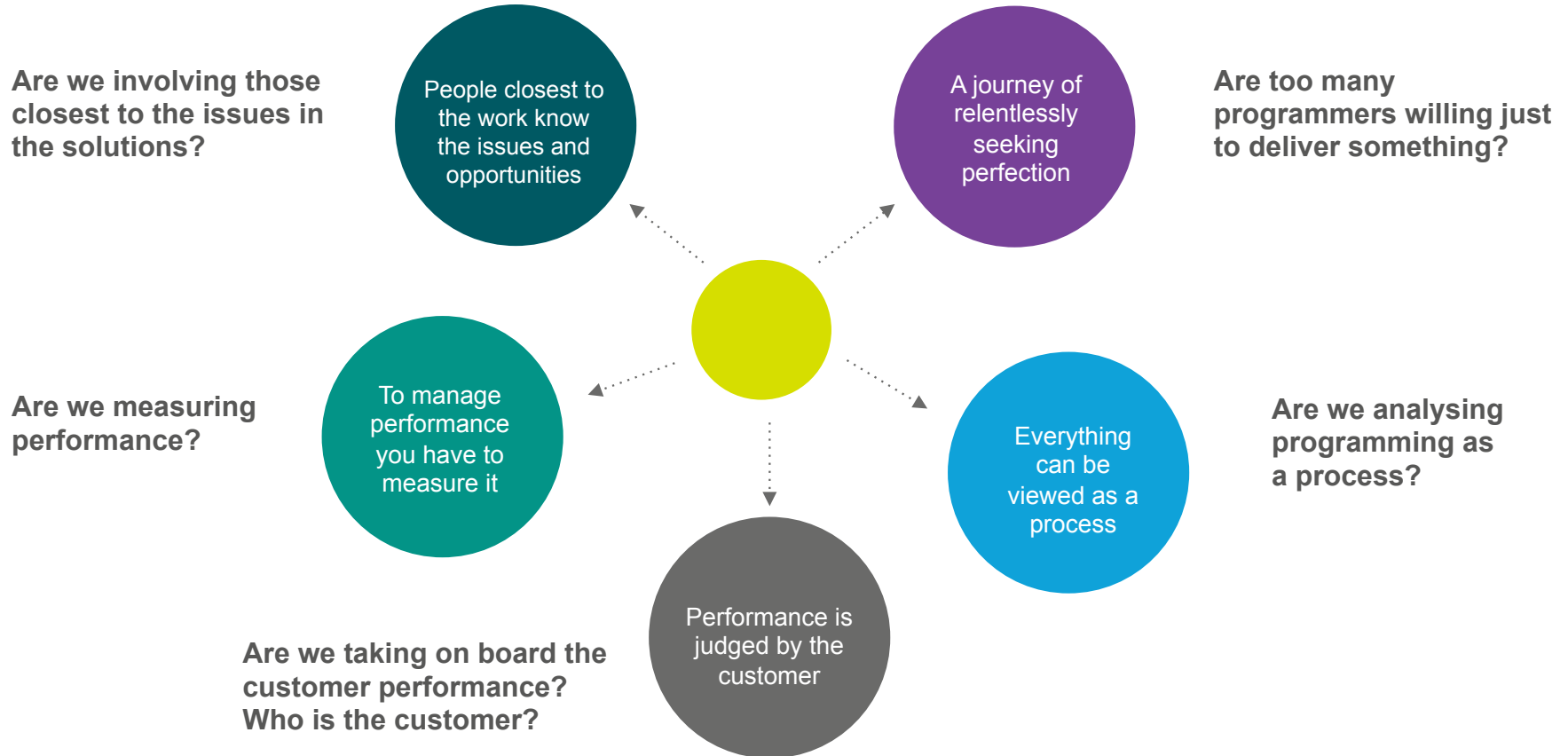
QUALITY



The 5 principles of continuous improvement



The 5 principles of continuous improvement



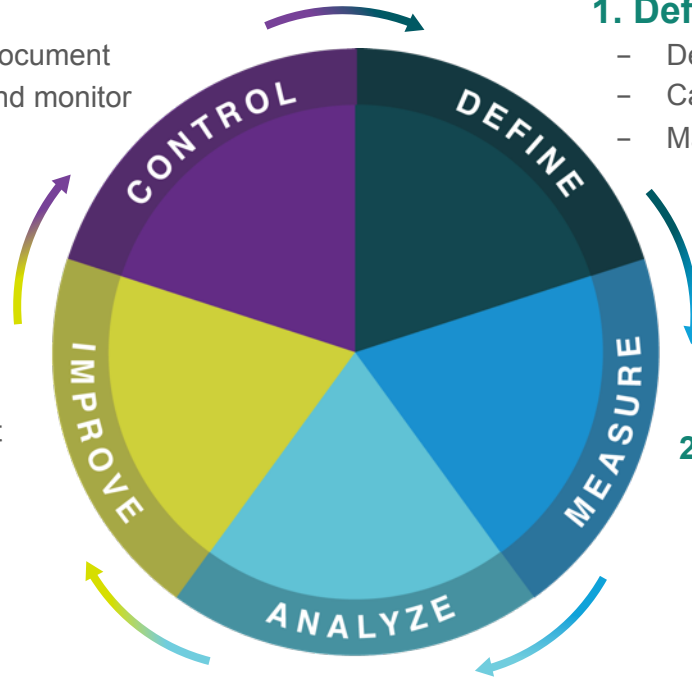
DMAIC – Continuous improvement's foundational problem-solving discipline

5. Control

- Standardize and document
- Evaluate results and monitor performance

4. Improve

- Generate and select solutions
- Pilot and implement solutions



1. Define

- Define the problem
- Capture Voice of Customer
- Map the existing process

2. Measure

- Gather and display data
- Time series plots
- Histograms
- Pareto charts

3. Analyze

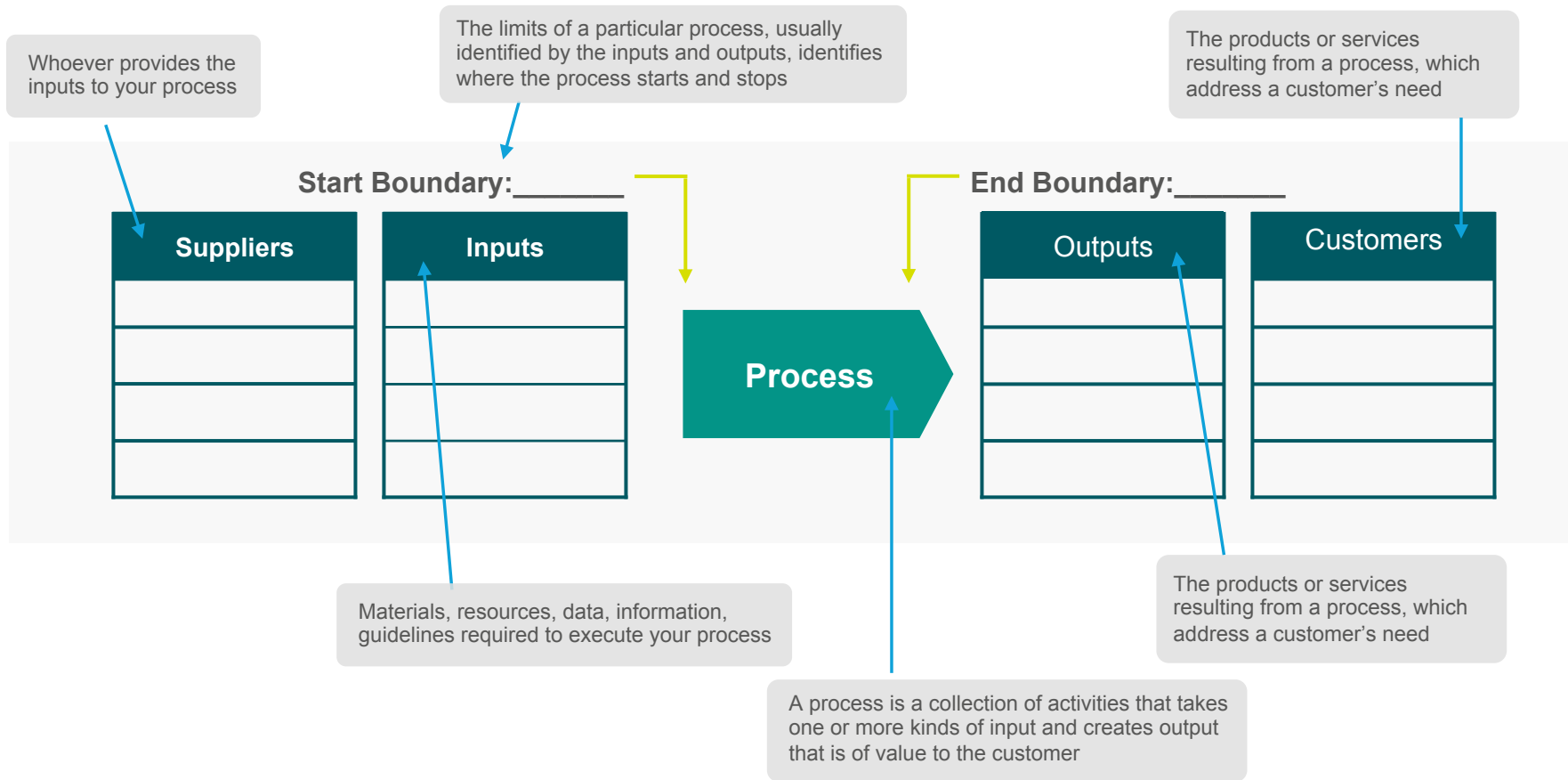
- Analyze the process
- Brainstorm root causes
- Analyze cause and effect

'Define' stage of process improvement

SIPOC is a process mapping tool used to help visualize the full scope of a process and what influences it, in terms of:

Suppliers
Inputs
Process
Outputs
Customers

SIPOC



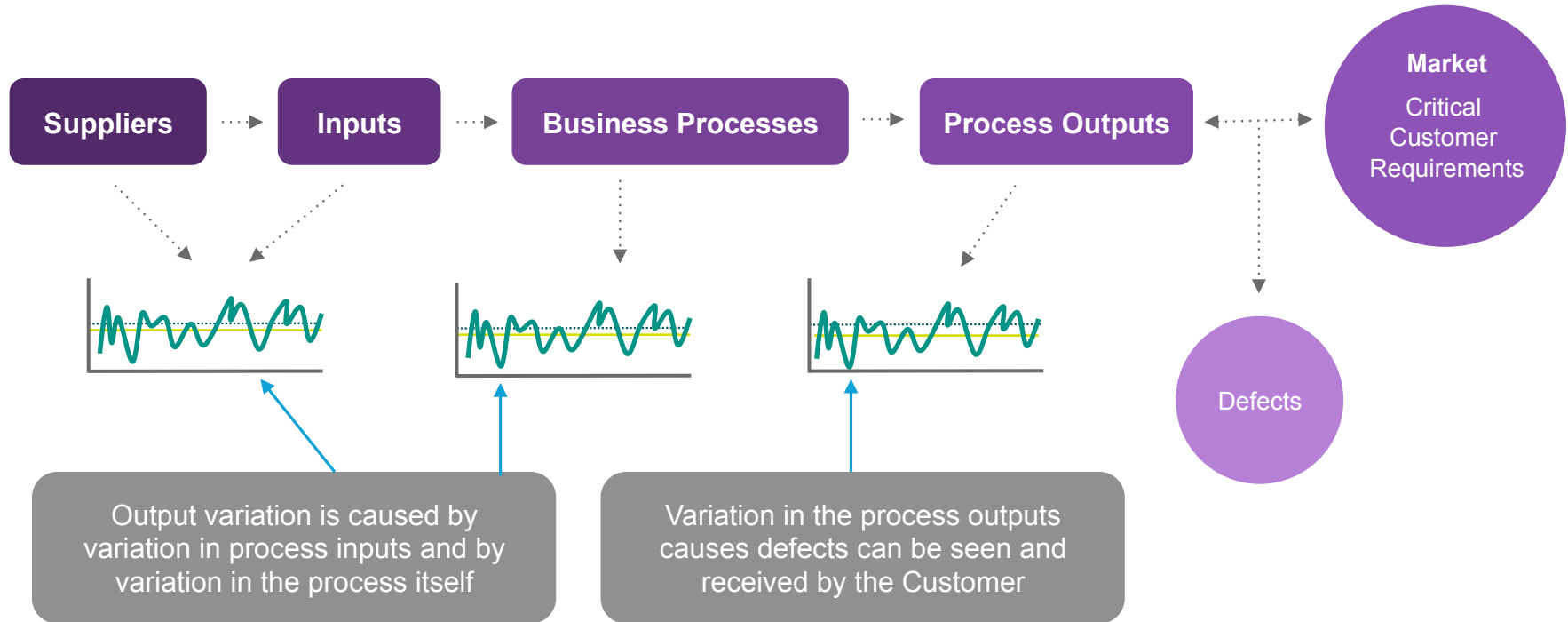
Defining the high level programming process

SIPOC – Supplier, Input, Process, Output and Customer

- **Suppliers** – DM, Clients, Biostatistics, Clinical, patients ...
- **Inputs** – Mock TFLs, Datasets, Protocol/SAP, rule books, conventions, previous studies....
- **Process** – Programming process creating output of value to the customer
- **Outputs** – TFLs, datasets, programs, logs, documentation.....
- **Customers** - Biostatistics, medical writing, clients, regulatory authorities, patients....

Focus on variation

SIPOC – Supplier, Input, Process, Output and Customer



Sources of variation in programming

Output variation is caused by variation in process inputs

Within Biostats and programming what do we do to control variability in inputs & suppliers?

Data variability:

- Std SDTM datasets / datasets mapped to std format
- Standard edit checks
- Coding of data ensures consistency
- Review of DM datasets
- Acceptance criteria decreases variability

TFL mock variability

- Standard mock templates
- Standard formatting agreements
- Detailed titles and footnotes

Sources of variation in programming

Output variation is caused by variation in process inputs

Relationships matter!

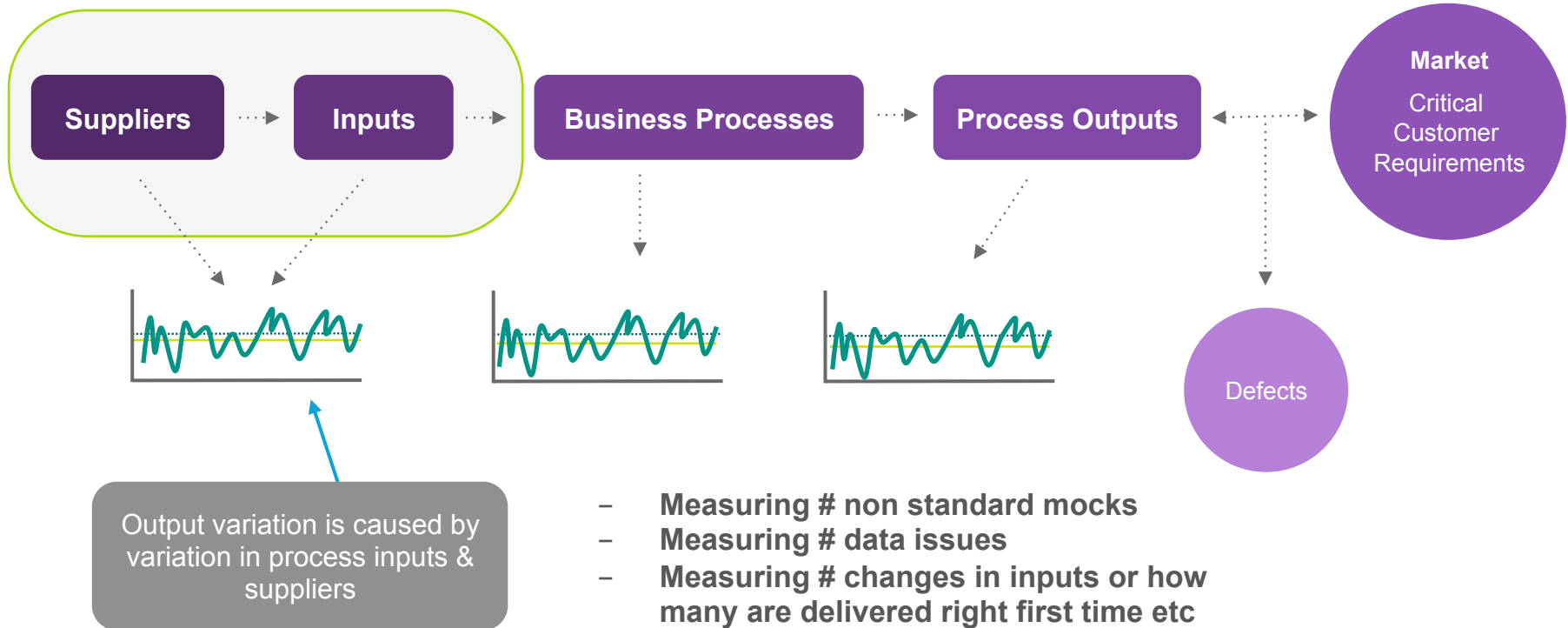
Process inputs may need to be discussed and clarified with the owner of the inputs

- e.g. DM – review data and provide feedback
- e.g. Biostatistics – clarification of specifications / analyses

etc

Measurement in variability of suppliers/inputs

SIPOC – Supplier, Input, Process, Output and Customer



Sources of variation in programming

Output variation caused by variation in the process itself

Programming process

- Map the process
- Investigate sources of variation
 - Geographical sources
 - Individual programmer variation
 - Therapeutic area variation
 - etc

Questions: Who should be involved with this review?

Remember: One of the 5 principles of continuous improvement

Sources of variation in programming

Output variation is caused by variation in process

Within Biostats and programming activities that control the variability in process?

Defining, documenting and explaining the process

This is why we all have mandatory SOPs /training

Repeatable process with minimal variations

Clear roles and responsibilities

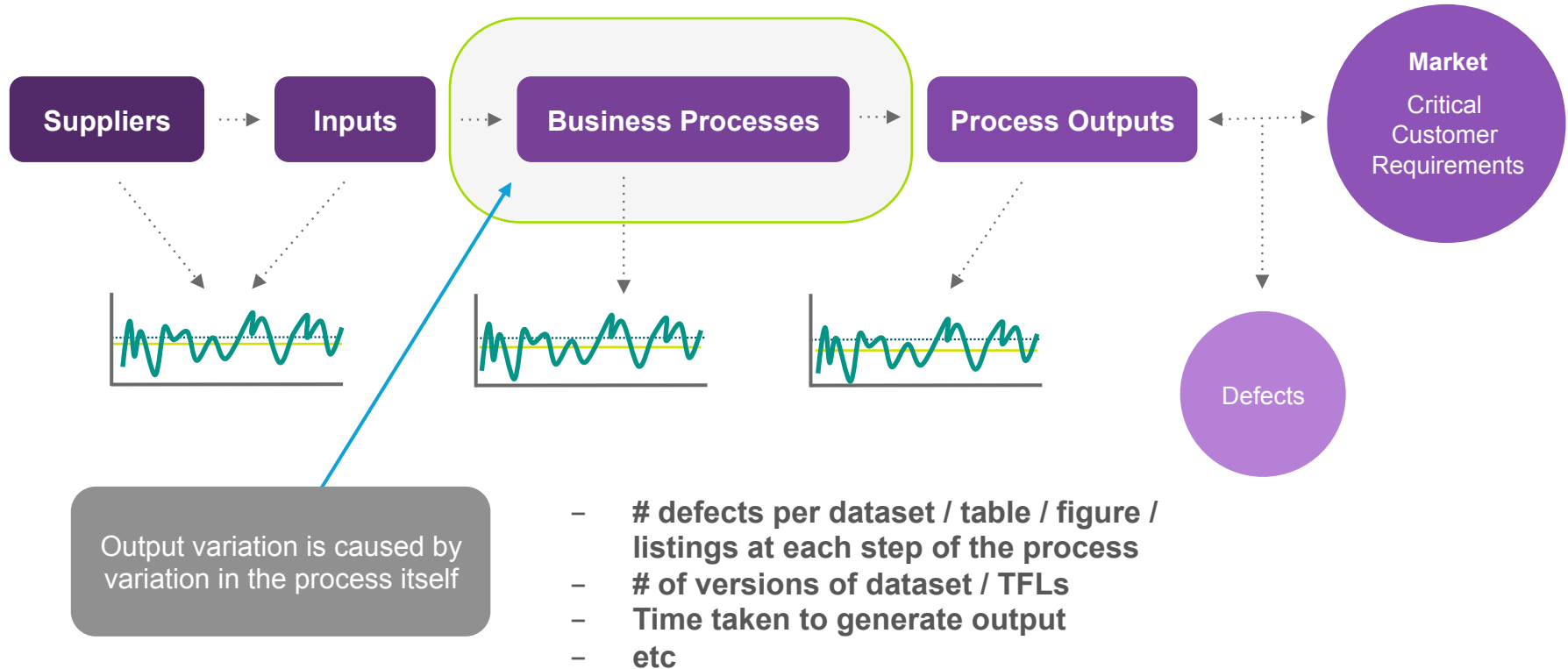
Validation programmer is responsible for final accuracy of content and format of the table

Standard macros / standard systems

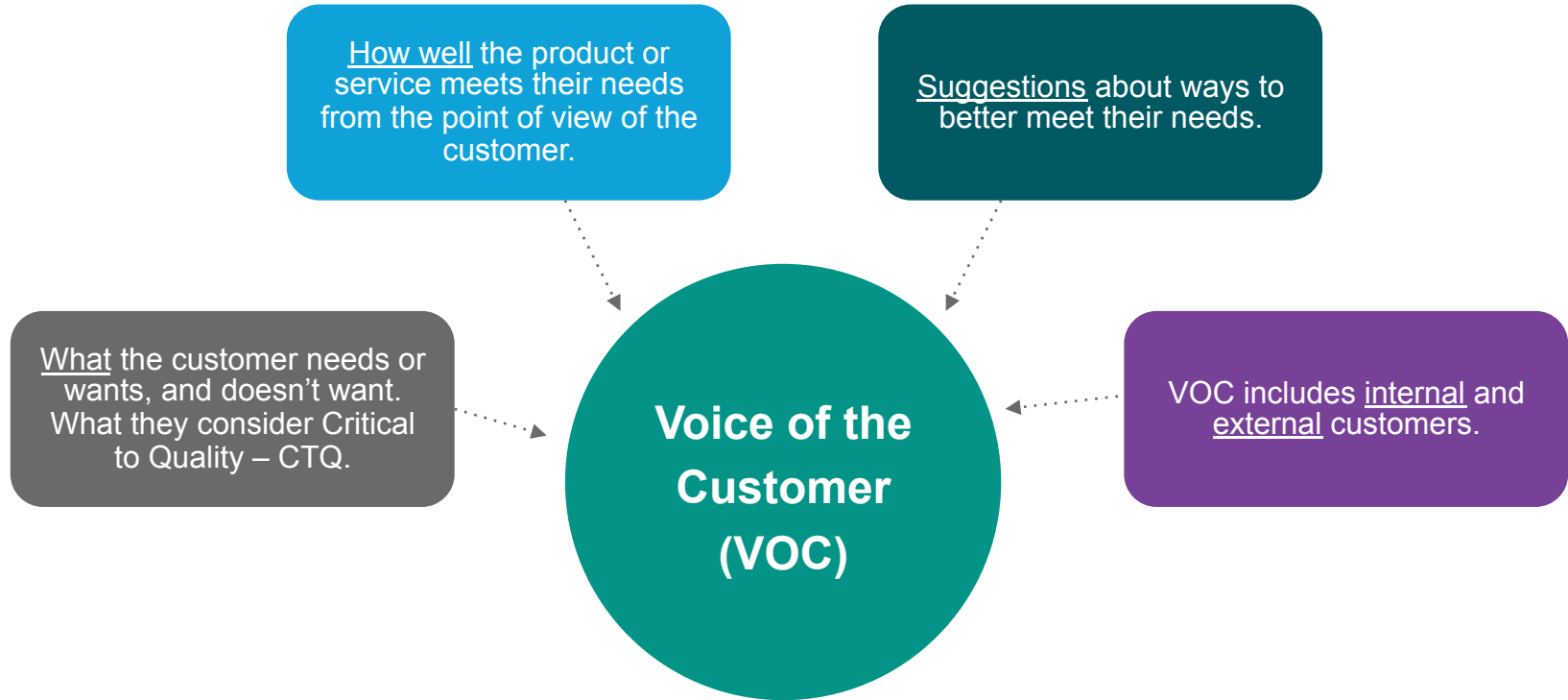
Robust coding practices

Best practice guidance documents

Measurement in variability of suppliers/inputs



Another concept from the 'Define' stage of DMAIC is the Voice of the Customer



Why is VOC important?

VOC data helps us to

Define what customers want:

- Product / service mix
- Product / service features
- Operating requirements

Define what customers don't want:

- Substandard performance
- Unwanted variation
- Waste

Prioritise and decide where to improve.

Identify specific solutions to explore

Two ways to obtain customer data



Passive Systems

Information comes to you whether you take action or not:

- Customer complaints
- Sales data
- Existing reports
- System alerts



Proactive Systems

You put effort into gathering the information:

- Ask the customer
- Brainstorm
- Focus groups
- Surveys

Common VOC mistakes

- Failing to probe for precise customer needs.
- Failing to probe for additional information: Why? How often? When?

Three features of effective customer questions:

1. Specific

- You said our outputs are poor quality. How do you define quality? What specific features do you look for?

2. Measurable

- You said that we are often late. What is late for you?
- You said we produce too many tables. What number do you want?

3. Prioritised

- You listed five features. Could you rank them?
- What are your non-negotiable requirements? . . . What else is important? . . . What would surprise and delight you?
- What do you like most? Least?

Measuring output variation

- **Defects in datasets / TFL deliverables to customers**
 - Major & minor
 - Acceptable error rate 0% for final delivery?
- **Defects in documentation**
- **Defects in cost**
- **Defects in timeliness**
- **Customer satisfaction**

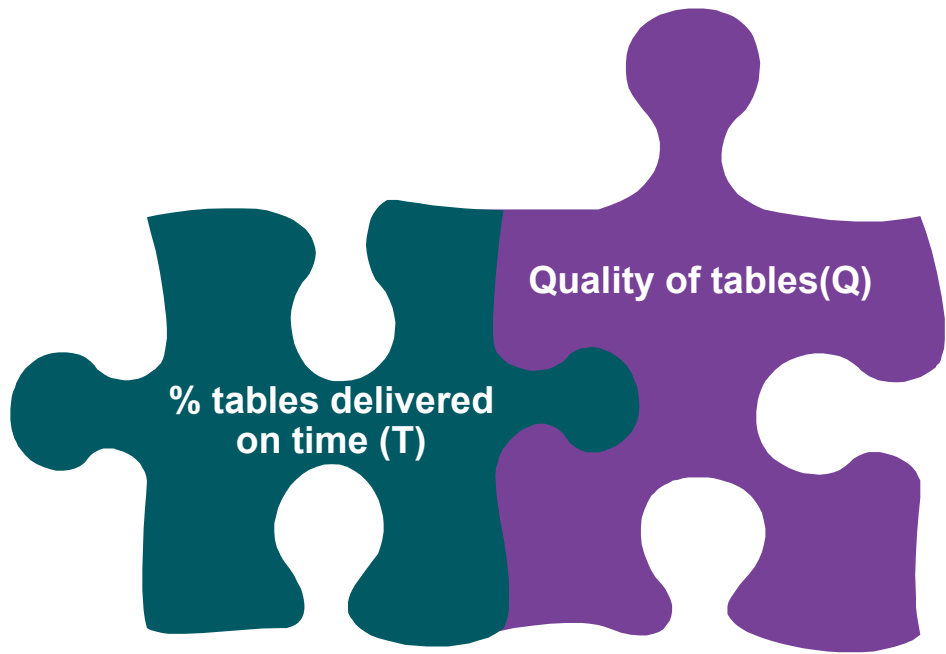
In programming in the past, focus on cycle times / timeliness

Alone each metric only tells part of the story ...



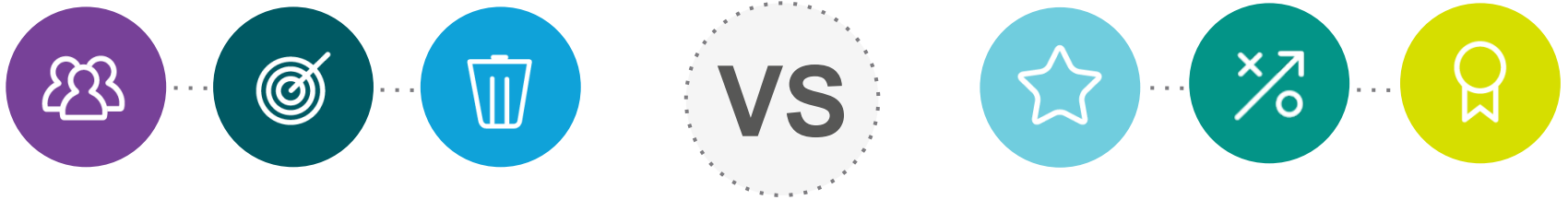
Idea of quality companion metrics

Reviewing these metrics together provides a more complete view of performance



Programming process: Challenge

Balancing expectations of the customer vs need to eliminate waste in the process



Programming process: Challenge



**Balancing expectations
of the customer**

VS

**Need to eliminate
waste in the process**

Programming process: Challenge

To manage performance/process step you have to measure it

Ideally would like to measure in automated fashion or easy to measure

Measure should be objective & specific

Each step of the process should have a performance measure associated with it

“Without data, you are just someone else with an opinion”

Code Review – Checks categorization

Type I

- Focusses on Program header, Modification history, Consistent indentation, comments, using of low case letters, spacing and number of characters per line in this category, which are easily adoptable

Type II

- Focusses on writing libnames, formats, writing sequential blocks, using data and out statements.. etc. these can be adoptable with little amount of efforts

Type III

- Focusses on usage of defensive coding, writing the complex program into simple blocks, checking logic of the program to ensure that it is correct, making sure that we are not losing data and hard coding will work for that extraction/ delivery only etc.

Some concluding thoughts

Tools from lean six can be used for programming process improvement

Lean six sigma reminds us of

- Importance of metrics
- Importance of voice of the customer
- Importance of trying to standardise and eliminate waste
- Planning vs fixing
- Defining what is critical to quality

PhUSE looking at

- Standard industry wide metrics

Thank you for your time.

ध यवाद्
dhan'yavāda

நன்றி
Nanri
Go raibh maith agat

iconplc.com

