Software Quality Metrics

Discuss the following items related to Software Quality Metrics

1. Software Quality vs Architecture Quality vs Project Quality
2. Characteristics of a Good Metric
3. Common Software Metrics
4. Software Metrics Best Practices & Recommendations
5. Measuring Output
6. Favorite Metric – Improve Weighted Average Defect Density
7. Example Metric Scorecard

Software Quality vs Architecture Quality vs Project Quality

This discussion is focused on the best was to measure the quality of an existing software application with the intent of improving the quality of the application. Other quality measurements include:

- Software Quality: The quality of an existing software product often measured by metrics like user satisfaction, maintenance costs, and defects.
- Architecture Quality: The quality of the underlying application design.
- Project Quality: The quality with which a given project was delivered often measured by metrics like on-time, on-budget, scope delivered vs committed, and cost.

Characteristics of a Good Metric

What makes a good software quality metric?

1. Easy to collect and understand
2. Good approximation of reality and good correlation to quality & goals
3. Meaningful to the group that can deliver positive change and can drive positive actions
4. Meaningful over time to encourage positive trends
5. Discourages “gaming the system,” or at least limits the impact of “gaming”
6. Does not allow one group to declare success (project management, developers, business analysts, etc.) while the overall product is failing
7. Comparable across teams, companies, and the industry

Common Metrics for Software Development

- Line of code per time-period
- On-time
- On-budget
- … Scope delivered
- Defects
- Weighted Defects
- Weighted Defects per unit delivered
- Unit delivered per hour/day/month… per person
- Unit delivered per hour/day/month… per dollar

Common Metrics for Agile Software Development

- Story Points per Sprint per person
- Say/Do ratio
- Story Points delivered per Sprint… change in Story Points delivered per Sprint
- Team versatility
- Percent coverage with automated testing or with Test-Driven Development
- Satisfaction surveys with internal or external customers
- Eric’s Favorite: Weighted Defect Density per 10,000 hours worked… production

Metrics Best Practices & Recommendations

1. Start small and focus on very few high-quality metrics
2. Distinguish between organizational level, team level, and team specific metrics
3. Distinguish between optional and mandatory metrics at each level
4. Promote and expand the usage of metrics that are working well within several teams
5. Promote optional metrics that are broadly adopted to become mandatory metrics
6. Regularly remove metrics that are not adding sufficient value
7. Insist on and reward transparency

Metrics Best Practices & Recommendations (continued)

8. Normalize metrics when appropriate
9. Consider surveys results when hard day is challenging
10. Consider using trends instead of absolute numbers
11. Consider productivity or quality improvement goals instead of absolutes
12. Never use metrics to punish a team
13. Reward teams for being transparent with painful metrics
14. Be very careful about metrics that attempt to measure productivity across teams

Measuring Team Output

We generally want to measure team output in order to:

1. Improve estimates
2. Improve output and velocity
3. Normalize metrics
4. Compare outputs across teams

Measuring Team Output

Three industry methods:

1. Story Points
2. Use Case Points
3. Function Points

Measuring Team Output to Compare Across Team

Be careful! No, be afraid!!

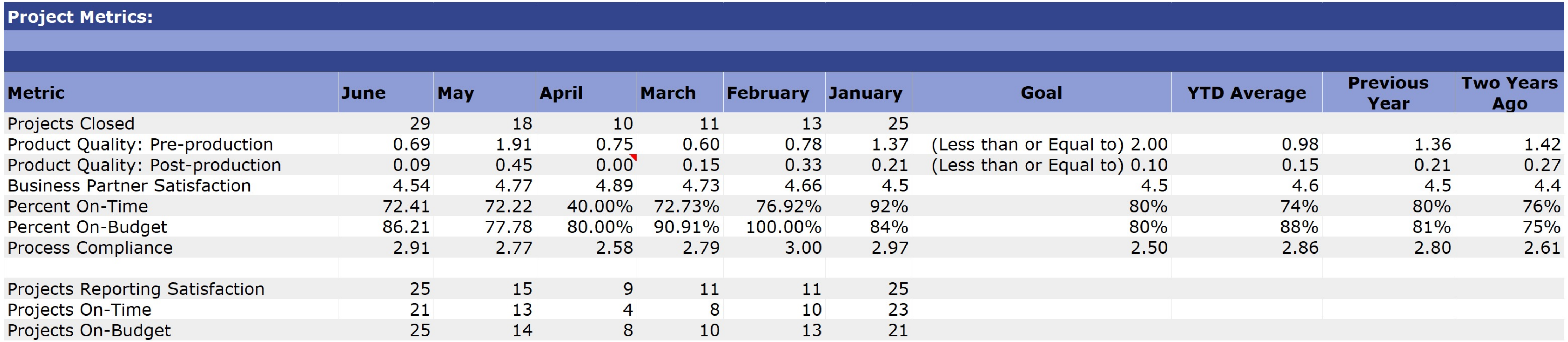
1. Story Points – can't utilize across teams
2. Use Case Points – might work in a RUP / Use Case enviornment
3. Function Points – painfully laborious, expensive, and often low confidence

Measure Output Across Teams & Good Metrics

What makes a good software quality metric?

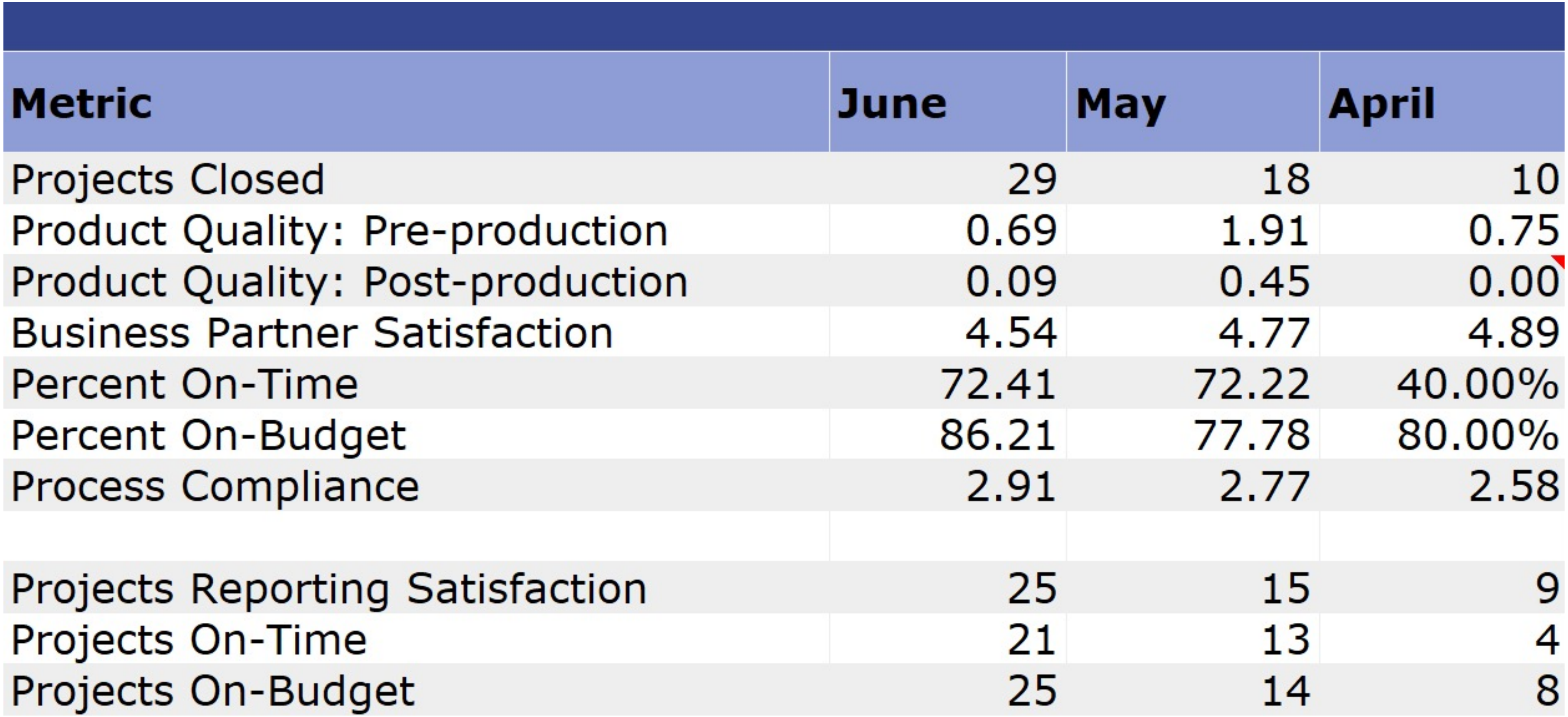
1. Easy to collect and understand
2. Good approximation of reality and good correlation to quality & goals
3. Meaningful to the group that can deliver positive change and can drive positive actions
4. Meaningful over time to encourage positive trends
5. Discourages “gaming the system,” or at least limits the impact of “gaming”
6. Does not allow one group to declare success (project management, developers, business analysts, etc.) while the overall product is failing
7. Comparable across teams, companies, and the industry

Example Metrics Scorecard



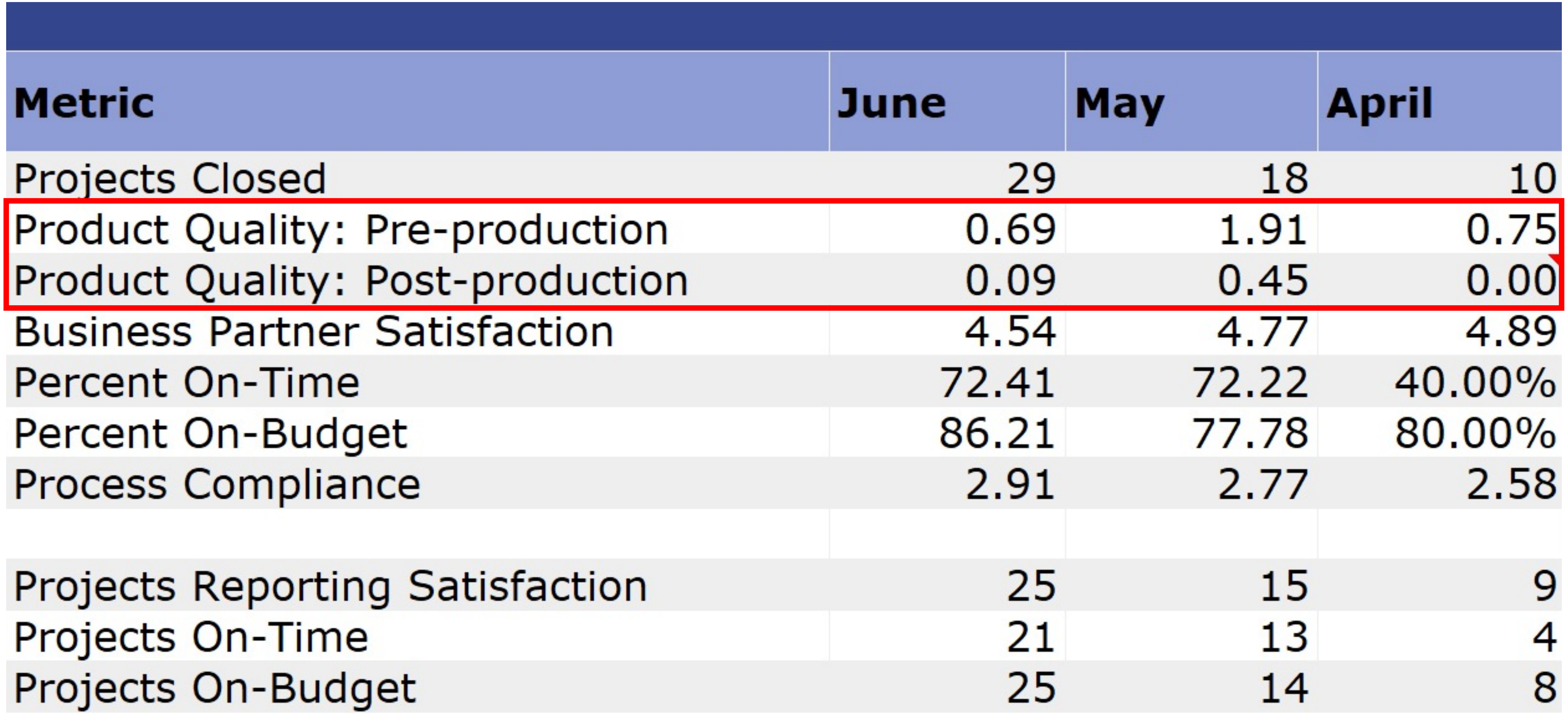
Project Metrics:

Metric	**June**	**May**	**April**	**March**	**February**	**January**	**Goal**	**YTD Average**	**Previous Year**	**Two Years Ago**
Projects Closed	29	18	10	11	13	25				
Product Quality: Pre-production	0.69	1.91	0.75	0.60	0.78	1.37	(Less than or Equal to) 2.00	0.98	1.36	1.42
Product Quality: Post-production	0.09	0.45	0.00	0.15	0.33	0.21	(Less than or Equal to) 0.10	0.15	0.21	0.27
Business Partner Satisfaction	4.54	4.77	4.89	4.73	4.66	4.5	4.5	4.6	4.5	4.4
Percent On-Time	72.41	72.22	40.00%	72.73%	76.92%	92%	80%	74%	80%	76%
Percent On-Budget	86.21	77.78	80.00%	90.91%	100.00%	84%	80%	88%	81%	75%
Process Compliance	2.91	2.77	2.58	2.79	3.00	2.97	2.50	2.86	2.80	2.61
Projects Reporting Satisfaction	25	15	9	11	11	25				
Projects On-Time	21	13	4	8	10	23				
Projects On-Budget	25	14	8	10	13	21				



Metric	June	May	April
Projects Closed	29	18	10
Product Quality: Pre-production	0.69	1.91	0.75
Product Quality: Post-production	0.09	0.45	0.00
Business Partner Satisfaction	4.54	4.77	4.89
Percent On-Time	72.41	72.22	40.00%
Percent On-Budget	86.21	77.78	80.00%
Process Compliance	2.91	2.77	2.58
Projects Reporting Satisfaction	25	15	9
Projects On-Time	21	13	4
Projects On-Budget	25	14	8

Weighted Average Defect Density



Metric	June	May	April
Projects Closed	29	18	10
Product Quality: Pre-production	0.69	1.91	0.75
Product Quality: Post-production	0.09	0.45	0.00
Business Partner Satisfaction	4.54	4.77	4.89
Percent On-Time	72.41	72.22	40.00%
Percent On-Budget	86.21	77.78	80.00%
Process Compliance	2.91	2.77	2.58
Projects Reporting Satisfaction	25	15	9
Projects On-Time	21	13	4
Projects On-Budget	25	14	8

Software Quality Metrics

Discuss the following items related to Software Engineering Metrics

1. Software Quality vs Architecture Quality vs Project Quality
2. Characteristics of a Good Software Quality Good
3. Common Software Metrics
4. Software Metrics Best Practices & Recommendations
5. Measuring Output to Normalizing Metrics and to Improve Estimates
6. Favorite Metric – Improve Weighted Average Defect Density
7. Example Metric Scorecard