Effective metrics provide an objective view of the work done in an organization. Metrics used effectively help teams understand their current state, identify opportunities for improvement, and self-manage their improvement cycles. Metrics used effectively at an organization level allow for well informed strategic decisions for investments, product development, and structure.

The determining factor here is the system optimization goal for the organization. If the system optimizing goal is secrecy then establishing silos where one part does not know the whole picture makes sense. In healthcare, I believe the goal should be to be high adaptability and high value. High adaptability enables us to effectively pivot to meet market demands, and high value ensures that we capture the market by delivering high value to the customers.

We discussed the need to evolve beyond measuring the basic table stakes i.e. team events, refinement, etc. In order to do this effectively, we as an organization need to identify where we are at and where we would like to be.

Here is a tool that offers some simple language to understand where we are in our agile journey and where we might want to get to as an organization.
https://www.leadingagile.com/our-compass/

My view would be that we are predictive-convergent (quadrant 2) headed towards being adaptive-convergent (quadrant 3).

Five steps for coming up with metrics:

1. Define the desired outcomes
2. Determine the key results that will lead to the desired outcomes
3. determine the metrics that should be used to measure the key results
4. determine where the data will come from
5. determine the frequency of the metrics data needed for insight

Themes as identified from discussion:

1. Planning Accuracy
2. Quality (stakeholder centric)
3. Scrum teams are highly efficient, performing, velocity (no waste, delays)
4. value being delivered (business adoption)
5. happiness index, sticky teams

Desired Outcomes:

Based on the conversation in the meeting, here are the desired outcomes that I think might help

1. Teams plan, coordinate, and deliver predictably enough to meet release level commitment
2. Teams deliver quality work to ensure customer satisfaction
3. There is nothing preventing the team from meeting its commitments
4. Team members are satisfied in their role/work, thereby reducing risk of team instability due to attrition

Ground Rules:

Based on the discussion, here are two ground rules that I kept at forefront:

1. What we measure tends to grow, therefore measure outcomes instead of output
2. Metrics are only as effective as their adoption. Metrics enforced as a stick give teams a choice between survival and growth. Survival wins.

Concepts at play:

As we solve the challenge of metrics in ORx, I think it is important that we come together on some concepts so that we can avoid being lost in semantics .

1. Measure is a data point at a single point in time. Eg: # of backlog items delivered in a sprint
2. Metric is a data point in context. Eg: # of backlog items delivered in sprint 3 compared to sprint 2
3. Leading metrics - are hard to measure, easy to influence and change Eg: Backlog Health
4. Lagging metrics - are easy to measure, harder to influence or change Eg: # of items delivered
5. A focus on measuring outcomes instead of outputs will create alignment within the organization. Eg: output would be how much stuff we deliver, outcome is how much value is delivered
6. Self-organized teams are those that own "how" the work is done along with "who" does the work
7. System Optimization: a system consists of whole/parts.
8. Local Optimization: Improving the parts individually often does not lead to an improvement of the whole.
9. Predictability vs Adaptability - balance the need for predictability with the need to rapidly respond to change. It's not so much about how predictable you are, it's more about how predictable you are trying to be. Conversely, if you value change and have built your organization for adaptability, we aren't so interested in how well you adapt, more how you have built mechanisms for adaptability into your organization. (https://www.leadingagile.com/our-compass/)
10. Convergent vs Emergent ecosystem - Are you trying to figure out your market, making sure that you are building the products your customer wants? We call that an emergent ecosystem. If you are focused on making and meeting commitments, possibly due to governance requirements or contractual obligations, you are probably operating in a convergent ecosystem. (https://www.leadingagile.com/our-compass/)

Metrics and Key Results:

Here is a list of metrics that I think would be useful in helping monitor the effectiveness of the team, product health, and organizational efficiency. My goal was to offer a list of metrics that we could sift through and in our discussions, identify those that apply the most with the ORx goal.

There are a lot of metrics below. The balance we are trying to strike is not to over-engineer and track all these metrics at once. Instead we as an organization need to discern which of these are important enough for us to actively monitor and which of these can we allow the teams autonomy to manage their continuous improvement cycles. This would mean that as leadership we expect the teams to be continuously improving and be able to articulate that progress when asked.

I think we think of these as 2 sets of metrics:

1. Metrics that teams can use for their continuous improvement
2. Metrics that would help us monitor the organization

Efficiency (Speed):

1. Lead Time
   a. Lead time measures how long it takes for a change to occur. This metric may be tracked beginning with idea initiation and continuing through deployment and production.
2. Deployment Frequency
   a. Deployment frequency denotes how often new features or capabilities are launched
3. Mean Time to Recovery
   a. Once failed deployments or changes are detected, how long does it take actually to address the problem and get back on track?
4. Mean Time to Failure
   a. determines how often deployments prompt outages or other issues.
5. Innovation rate
   a. the capacity of a team to build value-driven features versus non-discretionary work like defect fixes and application support.
6. On-product index
   a. measures the efficiency of how a team is run and how much they're contributing to a product's growth. By reducing unnecessary meetings and other distractions, you can increase a team's on-product index, which will "buy time" to work on valuable things.
7. Capacity planning
   a. leaving slack in capacity for unplanned work (in conjunction with UWR)

Planning & Delivery (Speed):

1. Velocity
2. Velocity by Work Type
   a. capacity invested in different types of work across the team (fixes, new development, product debt, etc)
3. Velocity Variance
   a. velocity variance from one sprint to the other
4. Story Points
   a. Committed to
   b. Completed
5. Unplanned Work Rate (UWR)
   a. The unplanned work rate (UWR) tracks this in relation to time spent on planned work. Ideally, the unplanned work rate (UWR) will not exceed 25 percent.

6. Throughput (instead of velocity alone)
    a. # of items delivered every sprint
    b. promotes smaller chunks of work being delivered
7. Deployment Time
    a. How long does it take to roll out deployments once they've been approved?
8. Cycle Time
    a. Shorter cycles are generally preferable, but not at the expense of discovering defects or abiding by SLAs.
9. Backlog Health
10. Cumulative Flow
11. Burnups/Burndowns

Quality:

1. Escaped Defects
    a. Escaped defects is a simple metric that counts the defects for a given release that were found after the release date.
2. Defect Volume
    a. used in conjunction with escaped defects. tracks actual volume of defects. While some defects are to be expected, sudden increases should spark concern
3. Defect Cycle Time
4. Cyclomatic Complexity & Static code analysis
    a. the code complexity metric counts a number of linearly independent paths through a program's source code. the more complex the code the higher the risk of compromised quality.
    b. Static code analysis tools help Agile teams check the structure of the code used to build a program, ensuring it adheres to established industry standards, such as indentation, inline comments, and correct use of spacing
5. Automated test coverage
    a. Automated test coverage provides a basic measure of risk to Agile teams. The more test coverage achieved by automation, the lower the chances of production defects in a release.
6. Code coverage
    a. Code coverage is a measure used to describe the degree to which code is exercised by your tests.
7. % of teams doing Continuous Integration/Test Driven Development (TDD)/Refactoring
    a. refactoring in every TDD cycle to design emergent code
8. % of teams that have installed telemetry
    a. Enable fast and continuous feedback from operations to development.
    b. Inserting telemetric data into your production application and environment, the DevOps team can automate feedback mechanisms while monitoring applications in real-time. DevOps teams use telemetry to see and solve problems as they occur, but this data can be useful to both technical and business users.
    c. telemetry data comes from logs, metrics and events. The measurements are things like memory consumption, CPU performance, and database response time.

Value:

1. Value delivered
    a. how much value did the customer receive? (mike T has a tool for calculating feature value)
2. Usage Index
    a. What features are people actually using? By analyzing the usage index of different product features, you're able to measure what's valuable to your users, an indispensable KPI for business value.
3. # of teams using Impact Maps (https://www.impactmapping.org/about.html)
    a. Impact mapping is a strategic planning technique. It prevents organizations from getting lost while building products and delivering projects, by clearly communicating assumptions, helping teams align their activities with overall business objectives and make better roadmap decisions.
    b. helps ensure that the right business outcomes are achieved, or that unrealistic projects are stopped before they cost too much, by clearly communicating underlying assumptions and allowing teams to test them.
    c. build objectives and key results into these impact maps

Team Morale (instead of happiness metric, see this):

a. I am enthusiastic about the work that I do for my team
b. I find the work that I do for my team of meaning and purpose
c. I am proud of the work that I do for my team
d. To me, the work that I do for my team is challenging
e. In my team, I feel bursting with energy
f. In my team, I feel fit and strong
g. In my team, I quickly recover from setbacks
h. In my team, I can keep going for a long time