OPTIMIZING FLOW IN THE SOFTWARE DELIVERY VALUE STREAM

The Ultimate Metrics to Become Predictable in Software Delivery

ConnectALL™

Throughput

What: The number of distinct pieces of user functionality (tickets, bugs, stories, etc.) delivered in a specific period of time, preferably weekly or biweekly.

Why: Predict the future or the done date. Plan out the value the team is capable of delivering.

Who: Agile Teams

Throughput Variation

What: Variation in throughput indicates an unpredictable value stream.

Why: Stabilize the team's performance. Manage risks and dependencies before starting the work.

Who: Agile Teams, DevOps Teams, Product Teams

Escaped Defects

What: Tracks defects that have escaped after the end of a sprint.

Why: Keep a check on the trends and see if your defects are decreasing. Understand how good of a job your team is doing as a whole. Know if you're sacrificing quality for speed.

Who: Agile Teams, Product Teams

Bottlenecks

What: Lets you know if there is a state in the workflow that is getting more work requests than it can process at its maximum throughput capacity.

Why: Pinpoints where changes can make the biggest impact in your software delivery value stream. Helps you understand where you need to focus to make the value stream more efficient.

Who: Agile Teams, Product Teams

Flow Load (WIP)

What: The WIP (work in progress) of a particular type on a particular day — the number of flow items being actively worked on in a value stream.

Why: Track inefficiencies caused due to too many flow items that are reducing output. Maximize Velocity and minimize Time

Who: Agile Teams, DevOps Teams

Distribution (Done)

What: Shows how much work your team completed in a time period and the specific types of work items that are being focused on.

Why: Understand how to accelerate delivery of business value by prioritizing on

Who: Agile Teams, DevOps Teams

Bug/Defect Aging and Defect Backlog

What: Shows if your team is neglecting high-priority defects in the process.

Why: Constantly check on the health of a software and improve customer satisfaction by ensuring major defects are fixed in a timely manner.

Who: Product Teams & DevOps

Deployment Frequency (DF)

What: Tracks the frequency with which increments of code are deployed to staging, testing and production.

Why: Ensures the delivery of valuable software 'early and often' by creating a trend towards more frequent deployments and small tickets.

Who: DevOps Teams

Mean Lead Time for Changes (MLT)

What: MLT is the time taken to go from code committed to code successfully running in production.

Why: Help your teams reduce their overall Lead Time and increase the velocity of software delivery. Be more predictable and accurate in determining when a change is expected to be delivered.

Who: DevOps Teams

Mean Time to Recover (MTTR) or Mean Lead Time to Recover

What: MTTR is the average time required to repair a failed component, system, service or defect that is in production.

Why: Track and manage build failure and reduce friction in deployment. Use this to manage recovery time and improve Deployment Frequency.

Who: DevOps Teams, Product Teams

80th Percentile Lead Time Expectation

What: Instead of the average lead time, the 80th percentile of your lead time indicates that 80% of the historical lead time falls below that point.

Why: Plan and make forecasts. Get a higher chance to over deliver by underpromising. Create a palatable customer experience by giving a 80% probability of giving a fix in a certain period of time.

Who: Agile Teams, DevOps Teams, Product Teams

- Business and/or Dev - Aligned to value, Business planning and funding
- Dev and/or Governance, Dev and/or Business - Velocity and Quality
- IT and/or Governance - Automation and predictability / Sustainability and Support
- Governance and/or IT - Risk mitigation and fast fix
- Product Teams