John Tanner
Agile Metrics – The GQM Approach
JOHN TANNER
SVP & Managing Consultant

John.Tanner@leadingagile.com
757.580.2078

www.linkedin.com/in/tannerjs
www.facebook.com/leadingagile
www.twitter.com/leadingagile

Married last Saturday to this beautiful woman...
DEFINING THE PROBLEM
THREE VIEWS ON METRICS

We need maximum metrics, because we just don’t know what we don’t know.

We need minimal metrics, because they just don’t add value.

We need some metrics, because something is better than nothing.
WE NEED ALL OF THE METRICS!

“Metrics are how we measure success. So let’s capture metrics for everything.”

Metrics dashboards become prolific, and someone has to maintain them.

Each Team winds up with a unique set of metrics to capture, but with no common expectation for using them.

The organization spends more time creating metrics roll-ups than they do creating product.

There is no cohesive strategy for tying the data together to make targeted improvements.
WE NEED NONE OF THE METRICS!

“Metrics do not add value, so they are waste. Let’s eliminate them altogether.”

Assumes all teams are currently producing working, tested product at high levels of efficiency.

Only works when the organization doesn’t care if they ever improve the system...

... doesn’t care if they aren’t producing value

... doesn’t get hung up on details like making payroll
WE NEED SOME METRICS...

“Metrics might help us improve, so we need measures of some sort.”

Without clarity, we often capture metrics for metrics’ sake

When we gather metrics without clear expectations around their use they can be wasteful, or even harmful

We focus on metrics in a way that ultimately drives “local optimization” at the expense of the overall system

We forget that the “why” and the “how” matters
THE “WHY” AND THE “HOW”
LET’S TALK ABOUT WHY, FIRST...
In Agile, we feel pretty confident we know how to measure success because...

The Team
... has a Stable Velocity
... is Making and Meeting Commitments
... is Producing Quality Software
TRADITIONAL APPROACH TO THE PROBLEM

- MAKES AND MEETS COMMITMENTS
  - METRIC: Story Point Completion %

- HAS STABLE VELOCITY
  - METRIC: Velocity Variance

- PRODUCES QUALITY SOFTWARE
  - METRIC: Escaped Defect Count
PROBLEM SOLVED?

Yay! We did it team!

No, but seriously, many people think like that...
HOW DOES THAT WORK AT SCALE?
WHAT IS GQM?

The Goal-Question-Metric approach is a simple model first developed by Dr. David Weiss, Victor Basili and the Software Engineering Laboratory at the NASA Goodard Space Flight Center.

In GQM we approach measures from three distinct levels:

- Conceptual level (Goal)
- Operational level (Question)
- Quantitative level (Metric)
WHAT IS GQM?

Conceptual level (Goal)

A goal is defined for a team or system, to meet a specific purpose, with a focus on demonstrating a meaningful outcome for the business.
WHAT IS GQM?

Operational level (Question)

A set of questions is crafted to define an objective model for assessing the achievement of our goals.
WHAT IS GQM?

Quantitative level (Metric)

A set of metrics, based upon the objective model, is associated with every question in order to answer it in a measurable way.
HOW DO WE USE GQM?

Identify the right goals for a team or system
Clarify our goals by asking the right questions
Identify measures and metrics that can answer the questions
Find ways to collect the data necessary to realize our measures and metrics
Analyze the data to assess whether we are answering our questions and achieving our goals
Use what we discover to improve our goals, questions, and metrics
EXAMPLE: MY GOAL FOR TODAY

Goal
My audience doesn't fall asleep in the next half hour

Question
Did they even show up in the first place?
Have they been thoroughly caffeinated?
Are they paying attention so far?

Metric
Empty Seat vs Sold Seat (Actual vs Expected)
Snore to Silence Ratio (Snore Complete / Silence Committed)
Eye Contact Variance
Heckling Density
EXAMPLE: ANOTHER GOAL FOR TODAY

Goal: Stay happily married

- Is she still here?
  - Metric: Wife in Seat Ratio (1:1)

- Is she angry with me right now?
  - Metric: Silence to Stare Ratio (+/- 40%)

- Are her needs being met?
  - Metric: Hangry Index (Minutes since last fed; 120 max)
REDEFINING THE PROBLEM
In Agile, we feel pretty confident we know how to measure success because...

The Team
  ... has a Stable Velocity
  ... is meeting Commitments
  ... is Producing Quality Software
In Enterprise Agile, we feel pretty confident we know how to measure success because...

Let’s figure it out, GQM style
GQM AREAS TO DISCUSS

Delivery Team Health (Scrum)
Delivery Quality
Technical Quality
Program Health
Product Quality
Portfolio Health
Portfolio Financials
GQM AT THE DELIVERY TIER
### DELIVERY TEAM HEALTH (SCRUM)

Teams can plan, coordinate, and deliver predictably enough to meet a release level commitment.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Question</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teams can plan, coordinate, and deliver predictably enough to meet a release level commitment</td>
<td>Does the team deliver the committed functionality each sprint?</td>
<td>Story Point Completion % (SPs Complete / SPs Committed)</td>
</tr>
<tr>
<td></td>
<td>Has the team established a stable velocity?</td>
<td>User Story Completion % (USs Complete / USs Committed)</td>
</tr>
<tr>
<td></td>
<td>Does the team have the resources it needs to be successful?</td>
<td>Velocity Variance</td>
</tr>
<tr>
<td></td>
<td>To what degree is the team encapsulated (i.e. make/meet commitments on its own)?</td>
<td>Team Stability Index</td>
</tr>
<tr>
<td></td>
<td>Is anything preventing the team from delivering on commitments?</td>
<td>% User Stories w/ outside dependencies</td>
</tr>
<tr>
<td></td>
<td>Does the team deliver throughout the sprint or at the “last minute”?</td>
<td>Blockers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sprint Completion Ratio (%SPs Accepted by Sprint End - 2)</td>
</tr>
</tbody>
</table>
Teams can frequently deliver working, tested software of a high level of quality.

**Delivery Quality**

<table>
<thead>
<tr>
<th>Goal</th>
<th>Question</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can the quality of the product being shipped be verified?</td>
<td>Open Defects</td>
<td></td>
</tr>
<tr>
<td>Are the teams able to address technical problems in a timely manner?</td>
<td>Defect Aging</td>
<td></td>
</tr>
<tr>
<td>Are technical problems getting past our quality checks?</td>
<td>Escaped Defects</td>
<td></td>
</tr>
<tr>
<td>Are technical problems being discovered in production?</td>
<td>Latent Defects</td>
<td></td>
</tr>
<tr>
<td>Is the team able to maintain the overall technical health of the product?</td>
<td>Technical Debt</td>
<td></td>
</tr>
</tbody>
</table>
Teams are able to predictably deliver working, tested software in a sustainable way.

**Goal**

- Do I have safety in refactoring?
- Can I easily understand and change the code in an expeditious way?
- Are we continuously improving our craftsmanship practices?
- Are we improving our ability to continuously deliver working software?

**Question**

- Code Coverage
- Cyclomatic Complexity
- Sonar Downtrend
- Build Cycle

**Metric**
GQM AT THE PROGRAM TIER
Program Management can make delivery commitments and successfully manage incoming requests.

- Is the roadmap of features sufficient to effectively maintain a ready backlog?
- Is the Ready Backlog sufficiently groomed to be predictably consumed by the DT?
- Is the Program effectively managing orchestration / mitigating dependencies?
- Is the Program (PO Team) focused on delivering value?
- Is the program delivering committed functionality each release?
- How long does it take for an Opportunity to move from Idea to Delivery?

**Goal**

**Question**

**Metric**

- Feature Roadmap Visibility
- % Ready Backlog
- % Ready Backlog User Stories w/ open dependencies
- % Value in Sprint
- Feature Completion Ratio
- Feature Lead Time
The organization delivers an intuitive experience and product features that delight our customers and produce the highest business value possible.

**Goal**

**Question**

1. Did the delivered solution meet the business need?
2. Does the Overall Product Delight the Customer?
3. Did the Feature Delight the Customer?

**Metric**

- **Product/Market Fit** (Actual vs Expected)
- **Net Promoter Score**
- **Pirate Metrics** (Acquisition, Activation, Retention, Revenue, Referral)
GQM AT THE PORTFOLIO TIER
PORTFOLIO HEALTH

- The organization delivers product features and capabilities with the best ability to deliver early ROI

<table>
<thead>
<tr>
<th>Goal</th>
<th>Question</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long does it take for an opportunity to move from “good idea” to “done”?</td>
<td>Epic Lead Time</td>
<td></td>
</tr>
<tr>
<td>Is the system delivering consistently?</td>
<td>Due Date Performance</td>
<td></td>
</tr>
<tr>
<td>Are we investing in the right type of work?</td>
<td>Portfolio Investment Mix</td>
<td></td>
</tr>
<tr>
<td>Are we focusing only on the most important things?</td>
<td>Value Stream WIP</td>
<td></td>
</tr>
<tr>
<td>Are we starting on too many things?</td>
<td>Abandoned Work</td>
<td></td>
</tr>
</tbody>
</table>
PORTFOLIO FINANCIALS

Goal

- The Portfolio is meeting its strategic, financial obligations and objectives

Question

- Are we getting the right return on our investments?
- Are we spending too much just to keep the lights on?
- Are we getting the expected results from this Portfolio?
- Are we meeting our targeted spend rates?
- Are we hitting our targeted value and maintenance blend?

Metric

- Value to Cost Ratio
- CAPEX to OPEX Ratio
- Variance to Results ($ or % Market Share)
- Variance to Spend Target
- Variance to Value/Maintenance Target ($ or % Spend)
REDEFINING THE PROBLEM (SOLVED)
MEASURE SUCCESS IN THE ENTERPRISE

The Portfolio

• ... consistently delivers product features & capabilities with a focus on time to market and early return on investment (Portfolio Health)

• ... consistently meets its strategic financial objects (Portfolio Financials)

The Program

• ... can make delivery commitments and successfully manage incoming requests (Program Health)

• ... can ensure the organization delivers the most effective experience and product features to delight our customers and produce the highest business value possible (Product Quality)

The Delivery Teams

• ... can plan, coordinate, and deliver predictably, in order to meet a release commitments (Team Health)

• ... can frequently deliver working tested code of a high level of quality (Technical Quality)
A CASE STUDY OF GQM FAILURE

- Introduced GQM to an existing team that had no measures in place
- Organization maturity was considered “low”
- No meaningful questions could be answered with existing data
- Productivity of the delivery team suffered
- Overall the implementation of GQM was considered a failure
- However, management noted they became clearer on goal setting

Source: A Software Measurement Case Study using GQM by Björn Lindström
CASE STUDIES IN GQM SUCCESS

• Two models for implementing GQM were introduced
  • Standalone GQM Team
    • Pros: Dedicated resource produced more consistent measures
    • Cons: Estimated 3 man-months of time spent per year on GQM alone
  • GQM Measures Captured by Delivery Teams
    • Pros: Measures are captured during normal work, team commitment to improvement
    • Cons: Other work can take priority over measurements and system improvements

CASE STUDIES IN GQM SUCCESS

• A hybrid model emerged
  • GQM team working with Delivery Teams to produce GQM measures

“In our opinion, ideally, the entire quality effort should be executed by the software developers themselves, being the most knowledgeable in their field. However, practice shows that in these cases most quality programs stop when project deadlines emerge.”

SO NOW WHAT?
NEXT STEPS

- Identify the right goals for a team or system
- Clarify our goals by asking the right questions
- Identify measures and metrics that can answer the questions

Find ways to collect the data necessary to realize our metrics and measures

Analyze the data to assess whether we are answering our questions and achieving our goals

Use what we discover to improve our goals, questions, and metrics
FINAL THOUGHTS
REFERENCES


“A Software Measurement Case Study using GQM” by Björn Lindström:

GQM Wikipedia Page:
https://en.wikipedia.org/wiki/GQM

Previous versions of this presentation:
https://www.leadingagile.com/2017/05/agile-metrics-gqm-approach/