Mariya Breyter
Beginning with an end in mind:
Re-Shaping an Enterprise with OKRs
Why am I here?

My specialty is transforming organizations, large and small, to a Lean and Agile mindset.

www.agileleantransformation.com
“The thing about goals is that living without them is a lot more fun...”  - Seth Godin

https://seths.blog/2009/01/the-thing-about/
Well-defined goals in an agile environment foster individuals’ motivation, ability to learn and grow, and impact their sense of meaningfulness by establishing ownership and accountability.

Team-level objectives enables teams’ collaboration and their ability to innovate and cross-pollinate. Team alignment empowers self-organizing teams.

OKRs are a goal-setting framework for thinking big. OKRs help establish high-level, measurable goals by establishing ambitious outcomes that can be tracked long-term. OKRs cascade across the enterprise, thus aligning all functions and creating transparent outcomes.
OKRs are based on the lean theory of system thinking and value delivery. This is an alignment and value delivery tool, not a performance assessment or productivity measurement tool. If used right, OKR framework supports alignment in value delivery and employee motivation across the enterprise. If used wrong, OKR lead to command-and-control methods, and loss of trust.

1. INDIVIDUAL MOTIVATION
   - What can go wrong?
     - OKRs used for performance assessments
     - Individual career aspirations not aligned with the role or with the team/management OKRs

2. TEAM-LEVEL GOAL SETTING
   - What can go wrong?
     - Top-down OKR assignment
     - OKRs are not aspirational
     - Key results are not measurable
     - OKRs not aligned with the program

3. PROGRAM-LEVEL DELIVERY
   - What can go wrong?
     - OKRs are “selfish” and optimize a component rather than the system
     - OKRs are not aligned with organizational priorities
     - No cascading OKRs

4. ORGANIZATIONAL ALIGNMENT
   - What can go wrong?
     - Unrealistic objectives
     - Budget allocation or bonuses based on OKR completion
     - OKR can be “gamed” to be easy to achieve

   What can go wrong?
   - OKRs are “selfish” and optimize a component rather than the system
   - OKRs are not aligned with organizational priorities
   - No cascading OKRs
A GAME

True or False?

1
Step 1. SET OKRs
GOOD or BAD OKRs?

OBJECTIVE - We will achieve market leadership for GitLab by
KEY RESULT 1 – Growing use of GitLab for all stages of the DevOps lifecycle by 10% via establishing three proven case studies (Product)
KEY RESULT 2 – Ensuring appropriate transactional business pricing (Sales)
KEY RESULT 3 – Launching advertising for the customer base for Manage, Plan, Create and increase overall pipeline coverage by 8% (Marketing)

*Source: https://about.gitlab.com/company/okrs/ty20-q2/#ceo-grow-incremental-acv-grow-website-visitors-introduce-a-certification-program-add-a-devops-transformation-offering
**GOOD or BAD OKRs?**

**OBJECTIVE** - We will increase efficiency of QA processes by:

**KEY RESULT 1** – Test cases for all P1, P2 stories are completed & handed over to dev before development starts (compliance to be measured every sprint)

**KEY RESULT 2** – 1 week before release date, no blocker & critical bugs should be open

**KEY RESULT 3** – Bug leakage to production for critical issues is less than 1%

**KEY RESULT 4** – Less than 3 bugs reported by end users per release

**OBJECTIVE** - Achieve market leadership for GitLab by

**KEY RESULT 1** – Growing 10% use of GitLab for all stages of the DevOps lifecycle via establishing three proven case studies (Product)

**KEY RESULT 2** – Ensuring appropriate transactional business pricing (Sales)

**KEY RESULT 3** – Launching advertising for the customer base for Manage, Plan, Create and increasing overall pipeline coverage by 8% (Marketing)
GOOD or BAD OKRs?

OBJECTIVE - We will deliver working software to the customer every sprint with high quality by
KEY RESULT 1 – Delivering to production every sprint (no single release is more than 1 day late)
KEY RESULT 2 - Achieving over 80% regression test automation
KEY RESULT 3 - Increasing unit test coverage to 75 % from current 45 %
KEY RESULT 4 - Implementing a continuous monitoring tool to ensure “six nines” uptime for lower environments
KEY RESULT 5- Enabling engineers manage lower environments within required SLAs

OBJECTIVE - Increase efficiency of QA processes
KEY RESULT 1 – Test cases for all P1, P2 stories are completed & handed over to dev before development starts (compliance to be measured every sprint)
KEY RESULT 2 – 1 week before release date, no blocker & critical bugs should be open
KEY RESULT 3 – Bug leakage to production for critical issues is less than 1%
KEY RESULT 4 – Less than 3 bugs reported by end users per release

*Source: https://upraise.io/objectives-key-results-okr/examples/engineering/
GOOD or BAD OKRs?

**OBJECTIVE** - We will achieve a higher operational availability and lower operational costs as measured by

**KEY RESULT 1** – Zero DevOps owned services in <Provider> (move everything to the cloud)

**KEY RESULT 2** - 10% reduction in <cloud implementation> operational cost

**KEY RESULT 3** – Zero single points of failure

**OBJECTIVE** - We will deliver working software to the customer every sprint with high quality by

**KEY RESULT 1** – Delivering to production every sprint (no single release is more than 1 day late)

**KEY RESULT 2** - Achieving over 80% regression test automation

**KEY RESULT 3** - Increasing unit test coverage to 75 % from current 45 %

**KEY RESULT 4** - Implementing a continuous monitoring tool to ensure “six nines” uptime for lower environments

**KEY RESULT 5**- Enabling engineers manage lower environments within required SLAs
OBJECTIVE - We will achieve a higher operational availability and lower operational costs as measured by
KEY RESULT 1 – Zero DevOps owned services in <Provider> (more everything to AWS)
KEY RESULT 2 – 10% reduction in AWS operational cost
KEY RESULT 3 – Zero single points of failure
GOOD or BAD OKRs?

OBJECTIVE - Achieve market leadership for GitLab
KEY RESULT 1 – Grow 10% use of GitLab for all stages of the DevOps lifecycle via establishing three proven case studies (Product)
KEY RESULT 2 – Ensure appropriate transactional business pricing (Sales)
KEY RESULT 3 – Launch advertising for the customer base for Manage, Plan, Create and increase overall pipeline coverage by 8 % (Marketing)

OBJECTIVE - Increase efficiency of QA processes
KEY RESULT 1 – Test cases for all P1, P2 stories are completed & handed over to dev before development starts (compliance to be measured every sprint)
KEY RESULT 2 – 1 week before release date, no blocker & critical bugs should be open
KEY RESULT 3 – Bug leakage to production for critical issues is less than 1%
KEY RESULT 4 – Less than 3 bugs reported by end users per release

OBJECTIVE - We will deliver working software to the customer every sprint with high quality by
KEY RESULT 1 – Delivering to production every sprint (no single release is more than 1 day late)
KEY RESULT 2 - Achieving over 80% regression test automation
KEY RESULT 3 - Increasing unit test coverage to 75 % from current 45 %
KEY RESULT 4 - Implementing a continuous monitoring tool to ensure “six nines” uptime for lower environments
KEY RESULT 5- Enabling engineers manage lower environments within required SLAs

OBJECTIVE - We will achieve a higher operational availability and lower operational costs as measured by
KEY RESULT 1 – Zero DevOps owned services in <Provider> (more everything to AWS)
KEY RESULT 2- 10% reduction in AWS operational cost
KEY RESULT 3 – Zero single points of failure
OKR WORKSHOP STEP 1: Define

Company OKR

**Annual Objective:**

---

**KR1:**

---

**KR2:**

---

**KR3:**

---

1. State your mission.

2. Formulate a compelling objective.

3. Split it into key results.

4. Share and get feedback.
Step 2. ALIGN
“Don’t limit your challenges. Challenge your limits.”

- Jerry Dunn
Sample Objective: **Improve Application Quality, Stability and Reliability**

**Key Result 1: Reduce number of production defects by 50%**

- QM: Reduce the number of missed defects to no more than 1 per 3 releases
- Dev: Ensure 100% knowledge transfer and SME on core systems
- Chief of Staff: Establish Production Support Team & define the process
- Agile Practice: Ensure that 20% + of sprint capacity is allocated to defect fixing

**Key Result 2: Improve quality of delivery within a sprint**

- Release Mgmt: Guide the process via reviews, reporting, and strategy definition
- Dev: Ensure 100% compliance with code quality standards
- Release Mgmt: Ensure that no stories with bugs open against them are deployed into production (exception: risk accepted stories)
- QM: Pilot TDD practices for 2+ teams
- DevOps: Environment setup allows for integration testing
- DevOps: Logging and Monitoring established with clear highly automated processes
- QM: 100% regression automation; 80% test automation

**Key Result 3: Increase System Stability and Reliability by 30%**

- Data: 99% data stability
- Data: 99.9% data accuracy
- Architecture: Proactive architecture definition and communication bi-weekly
- Security: Secure solutions defined, established and monthly communicated
- Agile Practice: Ensure that each backlog is balanced between functional user stories and non-functional technical tasks

- QM: 100% regression automation; 80% test automation
- Dev: Establish 100% coverage for peer reviews
- QM: 100% regression automation; 80% test automation
- Dev: Ensure 100% compliance with code quality standards
- DevOps: Environment setup allows for integration testing
- DevOps: Logging and Monitoring established with clear highly automated processes
Objective: Improve Customer Experience

Key Result 1: NPS score goes 2 points up
- Customer Service:
- IT: Delivery
- Agile Practice:
- Solution Architecture:

Key Result 2: Implement 10 highest priority features on a quarterly basis
- Customer Service:
- IT: Delivery
- Agile Practice:
- Solution Architecture:

Key Result 3: Establish a continuous customer feedback loop via three channels
- Customer Service:
- IT: Production Support
- Analytics:
- Security:
- Other functions?
OKR WORKSHOP STEP 2: Align

Objective: Improve Customer Experience

1. Select an alignment area.
2. Define your org structure and related OKRs.
3. Create a cascading view of your OKRs with clear ownership.
Step 3. REFINING
“Measure what is measurable and make measurable what is not.” - Galileo Galilei
**Company OKR: Respond to market needs by delivering new functionality to global customers daily**

**Objective: **We will enable daily production deployments to global customers with high quality at no additional expense by

**KR1: **Performing 80% of application testing without requiring an integrated environment.

**KR2: **Enabling daily deployments with full regression testing.

**KR3: **Reducing high priority and critical production defects to no more than one per 20 deployments.

<table>
<thead>
<tr>
<th>Objective</th>
<th>Key Results</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1: Improve application architecture to decouple solutions for testing and deployment purposes</td>
<td>Perform 80% of application testing without requiring an integrated environment</td>
<td>0-integrated environment is required in 100% of testing; 0.5 – 50%; 1 - integrated environment is required in 20% of testing</td>
</tr>
<tr>
<td></td>
<td>Deploy the application independently from other services/applications it depends on</td>
<td>0 – deployments are tightly coupled; 0.5 – deployments are loosely coupled; 1 – components can be deployed independently on demand</td>
</tr>
<tr>
<td>Q2: Ensure that all delivery teams are cross-functional</td>
<td>All delivery teams have skills necessary to design, develop, test, deploy, and operate the system on the same team</td>
<td>0 – none of the teams are cross-functional; 0.5 – 50% of teams, 1 - 100% of teams are cross-functional</td>
</tr>
<tr>
<td></td>
<td>Each team has access to manage relevant lower environments with a dedicated team member authorized to perform production deployments</td>
<td>0 – no access, all done by a dedicated team; 0.5 – lower environments only, 1 – all environments including production within agreed upon process</td>
</tr>
<tr>
<td>Q3: Ensure proper deployment tools and related skills</td>
<td>Optimize deployment tools and frameworks to enable daily deployments with full regression testing</td>
<td>0 – no changes; 0.5 – implement daily deployments with some manual testing; 1 - full regression automation</td>
</tr>
<tr>
<td></td>
<td>Ensure necessary skills are in place on each team</td>
<td>0 – no training; 1 - train everyone</td>
</tr>
<tr>
<td></td>
<td>Ensure all technologies are correctly licensed</td>
<td>0 – no audit; 1 - internal audit, 100% confirmed</td>
</tr>
<tr>
<td>Q4: Build quality in by implementing BDD for 100% of regression testing</td>
<td>Reduce high priority and critical production defects to no more than one per 20 deployments</td>
<td>0 – more than 5 defects per 20 deployments (current data), 0.5 – no more than 3 defects, 1 – no more than 1 defect</td>
</tr>
</tbody>
</table>
OKR WORKSHOP STEP 3: Refine

1. For each of your objectives, create quarterly KRs.

2. Come up with pre-grading for each KR.

3. For each external dependency, get a handshake. If not possible, refine until it is achievable.

4. Agree on ownership.

**Annual Objective:**

KR1:

KR2:

KR3:

<table>
<thead>
<tr>
<th>Q1 Objective</th>
<th>Q1 Key Results</th>
<th>Pre-grading/Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 4. ITERATE
“Continuous improvement is better than delayed perfection.” - Mark Twain
Sample Annual OKR Cadence

- Early Nov.
  - Announce Enterprise OKRs
  - Divisional OKR Drafting
  - Alignment Workshops

- Mid-Nov.
  - Review the grading of last year’s OKRs
  - OKR Kick Off
  - Team OKR drafting and alignment
  - OKR Retrospective

  - Q1 grading and refinement
  - Q2 grading and refinement
  - Q3 grading and refinement

- Early Dec.
  - Q4 grading and refinement
SUMMARY: 4-STEP OKR PROCESS

1. Set up
   - Q1 Objective
   - Q1 Key Results
   - Score

2. Align
   - Review and grading of last year’s OKRs
   - OKR Kick Off

3. Refine
   - Team OKR drafting and alignment
   - OKR Retreat
   - Divisional OKR Drafting
   - Alignment Workshops

4. Iterate
   - Announce Enterprise OKRs
   - Early Dec.
   - Mid-Nov.
   - Early Nov.

Objective
- Improve Application Quality, Stability, and Reliability
- Key Result 1: Reduce number of production defects by 50%
- Key Result 2: Improve quality of delivery within a sprint
- Key Result 3: Increase system stability and reliability by 30%
OKR Manifesto

“The thing about goals is that living without them is a lot more fun, in the short run.”

“Don’t limit your challenges. Challenge your limits.”

“Measure what is measurable and make measurable what is not.”

“Continuous improvement is better than delayed perfection.”
THE GAME

1. True or False?
OKR BIGGEST SECRET

“The greater danger for most of us lies not in setting our aim too high and falling short; but in setting our aim too low, and achieving our mark.”

- Michelangelo

Bring OKRs home!
You can find me at:
https://www.linkedin.com/in/mariyabreyter/
www.agileleantransformation.com
https://www.100careertips.com/
maria_breiter@yahoo.com