



Minimum Viable Agile

Your starting point in Agile Governance

Overview

Why an MVA?

We enable different teams to shape their own ways of working to be as effective as possible while working towards our vision and mission. To ensure that we don't end up in chaos we agreed on a common baseline – the MVA.

What is the MVA?

The MVA creates a baseline on how we work together and represents a minimum set of principles, practices, processes and tools which help us in taking ownership, speed up and streamline the way we collaborate with each other and manage dependencies.

How to inspect & adapt the MVA?

Every quarter there will be a Session, where representatives of every organizational unit participate and provide input around the MVA (this document) on the things we think we should add, change or remove.



MVA

How we prioritize our work

We have different levels of priorities at our organization, which also can influence one another:

1. **Disruptions:** Incidents on production that are critical to be resolved asap independent of the time and place they occur, meaning that incidents can also occur while being on call. Health metrics help us to keep our products and system healthy to ensure we can prevent them from failing.
2. **OKRs:** are our main priority and focus. They are defined top-down (with bottom-up input).
3. **Business as usual:** small operational work we need to do that is not business critical, however can unblock other departments or enable stakeholders to be more effective.

How we organize our work

Within our organization we have a shared understanding on how we organize ourselves in terms of structuring our work, to make sure we are as transparent as possible with all our stakeholders.

Agile Work Breakdown Structure (WBS)

Reflects the different levels of our product backlog. With the agile WBS we increase transparency (as everyone knows what structure to expect), have an easier time



aligning both on a business and technical level and collaboration can happen more easily.

High Level (less details)

- **Initiatives:** reflect hypotheses we are betting on to reach defined OKRs or within a specific Program.
- **Epics:** Can be sub-parts of initiatives and describe a part of the initiative that provides user or customer value. Epics normally include a group of items with a common business goal being discussed amongst Product Owners and Product Managers.

Low Level (more details)

- **User Stories:** are normally broken down from Epics and reflect small increments of our product and/or services that deliver user or customer value. Issues found within a user story that is not publicly available yet, need to be fixed within the user story (so no incident or bug is created).
- **Bugs:** reflect functionality that is not working properly as expected which already was made available for our users and customers (production). Bugs don't have an immediate critical impact on our business and so can be prioritized accordingly.
- **Incidents:** an identified event and/or issue, which causes an immediate (negative) impact on our business and/or technology, unplanned interruption or reduction of quality of the service.
- **Spikes:** an item for investigation, things we need to figure out, decisions we need to make. In other words: it's an effort to learn something and make a decision.
- **Sub-Tasks:** are normally broken down from (low level) product backlog items such as user stories, spikes etc. and describe the specific way, which the team



agreed upon, to achieve a user story.

How we plan and deliver work

Within our organization, we align first on OKRs giving us the direction of what impact we want to achieve. With that information we come up with Initiatives we are betting on, to fulfill the OKRs.

The Initiatives are broken down based on our Agile WBS and transferred into a high level roadmap.

High Level Roadmap

A high-level roadmap at our organization shows in a rough (via Epics) way, which parts of the entire initiative we think are able to deliver in which order and in what timeframe. As we live in a constantly evolving and changing environment, the high level roadmap is a living document and always reflects the actual and current state.

Agile Estimations

Estimations help us to have a rough idea on how complex the problems are that we want to solve for our users and customers. They also help us in prioritizing, communicating and managing expectations with stakeholders, so that they know if it makes sense to invest in the proposed initiatives, epics, user stories and/or even bugs.

We use probabilistic forecasting based on historical data to continuously forecast our initiatives. If an Initiative is approved the responsible Team(s) will start breaking



the work down to be able deliver the most valuable parts of the Initiative to our users and customers. Here we agreed on having a specific rhythm where we deliver small product increments, which we call cadence.

Cadence

Cadence provides a steady rhythm to our delivery process. It helps us to:

- improve predictability, planning, and coordination
- align & control injection of new work across the organization (as this happens at a specific time and place)
- make waiting times more predictable
- and much more

To gain the benefits of cadence teams implement a rhythm to allow them to remain aligned within the organization or its units (departments, teams, etc.). The teams define what rhythm is helpful for them as long as it doesn't negatively impact the organization or its units.

This means that activities that require coordination or are needed as a base for other activities need to happen either in sync (e.g on the same week, day and time) or timely before the dependent activities start.



The **only exceptions** where the cadence is defined are:

- Operations Review (every Month): is a meeting to analyze the efficiency of our set-up. Representatives from all teams share their findings that concern the organization-level changes, define ways to measure improvements and agree and commit on ways to improve.
- Service Delivery Review or Retrospectives (every 2 weeks): the team focuses on improving their ways of working and sdlc based on available metrics.
- Show and Tell (company wide, every Month)

Within the defined cadence teams follow their own cadences, to deliver value for our users and customers.

Classes of Service

At our organization we have the following Classes of Service: Expedite, Standard and Intangible with each of them having either a capacity limit (either in number of PBIs, % of capacity or SP).

Expedite class of service is used for critical and top priority items that require immediate handling. An expedite item preempts other work and requires teams to ensure no interruptions are experienced in the workflow. Some examples are: Incidents on production, PBIs that if not done timely we lose money, etc.

Standard is used for everything that is related to an ongoing initiative, epic or OKR. Normally items in this class of service aim to solve a business, user or customer requirement.



Intangibles cover everything that doesn't have a high impact right now, which however can become critical if not solved within a mid-term timeframe. Some examples are: QA automation, small tech debt, etc.

Definition of Ready (Baseline)

The Definition of Ready (DoR), gives us an overview of when a User Story or Bug is deemed ready to be worked on by a Team.

Each Team should define their own DoR however the criterias described below must be included:

- provides value for users/customers.
- is refined, estimated, known and understood by each team member
- is testable (in terms of UAT, QA-Automation, ...).
- is small enough to be delivered in under 1 Weeks.
- follow our templates
- Follow INVEST criterias (Independent, Negotiable, Valuable, Estimable, Sized or Small, Testable).

Definition of Done (Baseline)

The Definition of Done (DoD), defines the criterias that must be met, so that a user story or bug can be regarded as done. Each Team should define their own DoD however the criterias described below must be included:

- code was reviewed by a second Engineer.
- passed all the functional and non-functional tests (manual, automation, SonarCube, etc.).
- merged to the main branch and deployed at least to staging.
- documentation updated to reflect current state (API Endpoints).
- analytics events are set and tracked.



How we collaborate

Continuous Collaboration

Each Team needs to have scheduled events in place where they reserve time to be able to collaborate frequently, enabling them to plan, forecast, align, adapt & inspect continuously. The Team is responsible to have every event in place that is needed to achieve the mentioned goals.

Examples of some Events:

- Scrum Events: Sprint Planning, Refinement, Daily, Review and Retro
- Joint Events: Joint Review, Joint Retrospective, Product Planning

Collaborating across teams

Sometimes teams will need to work together on one Initiative / Epic / User Story.

Here alignment and working together are key, so the Teams involved need to think on how to solve the alignment.

Here you can find some practices which worked well in different environments in the past:

- Implement code in a code base they don't own, however the team that owns the service needs to review the changes
- Ask the service owning team to implement the changes for them, however keep in mind that this approach creates dependencies, and the requesting team needs to check if this delays their plans
- Do Scrum of Scrums (SoS) sessions regularly to improve collaboration between the teams and enable to resolve dependencies on a short term
- Pair programming for a specific PBI

Architecture decisions which have dependencies over multiple teams need to be worked on by the teams involved and proposed to the architecture group.



How we continuously improve

At our organization we know that continuous improvement is key, to be successful. That's why the teams are responsible for taking action on how to improve their processes, technology and skills. To support the teams in this endeavor, we have a couple of helpful events, tools and methods in place which are described below.

Communities of Practices

A community of practice (CoP) consists of a group of people who are interested in a specific discipline within our organization and want to gain and share knowledge around that discipline. Everyone, independent of their level of expertise, is welcomed in a CoP. A CoP is self-organized, meaning that topics that are discussed are decided collaboratively by the members of the CoP.

Team Health Checks

Team Health Checks give the Team visibility on where they excel and which areas need improvement. Based on the Team Health Checks, the Teams can take specific actions on how to improve.

Note: Team Health will be covered during the Team's retrospectives in regular intervals, however at least once a quarter.

How we measure success

Working data driven is a discipline within our organization that we want to pursue. We want to measure how successful we are and spot the need for improvement to steer accordingly towards success.



At our organization we defined three key areas that we want to focus on:

Achievements

This part covers everything related to our achievements in terms of supporting other teams and departments as well as having an impact on the business.

Example of metrics in this area are:

- OKRs
- Fulfillment of SLAs and/or SLEs (KANBAN Teams)
- Sprint Goals achievement rate (SCRUM Teams)