

Exploring Outcome Based Agile Contracts

Q1. What is the difference between an Outcome based contract (OBC) vs. Standard one?

Some of the key differences are mentioned here:

Area	Outcome Based Contracts	Standard Contracts
Objective	Based on results, quality, value achieved by the customer	Based on inputs like scope, time, cost
Price	Depends on performance, specific business outcomes	Depends on either the scope of work, time estimated or cost involved like headcount
Risk Management	Shared between customer and supplier	Borne either by the customer or the supplier
Transparency	Processes, tools, data and means to achieve outcomes	Only results are made available

Q2. What is changing from respectively the customer and vendor standpoints in OBC?

This how the change is perceived by the customer and the supplier:

Area	Customer Standpoint	Supplier Standpoint
Partnership	Must be able to see the vendor/supplier as a collaborative partner , hence includes the vendor/supplier/provider in strategy building. The partnership is therefore, more engaging	<ul style="list-style-type: none">• The vendor/supplier changes their mode from mere supplier of resource/people/technology to a collaborator.• The vendor/supplier/provider provides their best input in terms of expertise, knowledge and resources to partner with the customer
Planning	Must involve the vendor/supplier early for strategic planning	Participates in framing the structure of the contract as they are invested in achieving customer's goals
Metrics	Relies on outcome based metrics than inputs such as agreement to evaluate vendor/supplier merit	Provides access to all relevant data that could help the customer: <ul style="list-style-type: none">• perform data analytics• generate meaningful metrics
Risks	Defines risks and expectations while also trusting the vendor for quality deliverables	Shares the risks of not achieving the goals in time. That is, if the outcomes are not achieved in time, the supplier payment might be impacted

Q3. What is the difference in work estimation process with OBC and who has final word?

Work Estimation in Standard Contracts

Fixed Price/Scope/Time:

1. Requirements are estimated in days, weeks and months
2. Requirements are split into WBS
3. WBS-based estimation is most sought by teams, in Hours and Days
4. The WBS-items are estimated by team members in groups,
5. Lot of assumptions are made as estimation to be provided upfront
6. Risk assessment and contingency planning needs to be done
7. Objectives are not very clear, uncertainties are relatively higher
8. Quality is impacted

Work Estimation in Outcome Based Contracts

"Focus on small stories that are thin vertical slices through the technology stack."

Norton, Doc. *Escape Velocity: Better Metrics for Agile Teams* (p. 60). OnBelay Consulting, LLC. Kindle Edition." (refer to Image 1)

1. First set of outcome(s) is(are) envisioned by the Customer and Supplier collaboratively. These are agreed to be the first level of MVP
2. First MVP is defined. A typical MVP size is not more than 20% of all expected deliverables.
3. The MVP is estimated in size and complexity points(Story Points)
4. The MVP is split into smaller requirements.
5. This is taken by the team of experts from the Supplier side, or the ones who have the right skill sets
6. The MVP is structured to get required level of requirements which are then picked up the team members
7. The estimations are done together by the team members, different requirements are relatively sized
8. For providing a date, the requirements are split into lowest level activities (Sub-tasks), each Sub-task estimated in Hours and/or Days and finally summed up into individual requirements to give a duration.
9. This duration provides an estimate of date of availability of the MVP
10. One or more of these MVPs are combined into getting a MMP or an Outcome
11. For each successive MMP, same steps are followed.

Q4. What KPIs are not making sense anymore in OBC?

Any KPIS/ metric used as is will not make sense in OBC.

Let us take some examples:

- **Velocity:** Velocity is calculated as the rate at which value is delivered to the customer.
 - There are some challenges associated with Velocity as a metric. Some of these are:
 - Associated with completion of Stories. The fact that whether these Stories are in Production or are implemented at the customer end, are not considered.
 - No indication of improvement of the process or are just delivering with same set of processes
 - A stable or slightly increasing velocity may be factor or known and predictable work than performance of the teams or the value delivered to the customer
- **Throughput:** Throughput is calculated as the total number of issues "Done" in a cadence or any unit of time.
 - The challenges associated with Throughput as a metric are:
 - Throughput considers bugs and technical tasks along with the asks from the customer
 - Throughput will consider every minor change in the requirement as a number

Likewise there are other metrics that directly do not contribute to the OBC

Some of these are:

- **Defects per release** = Escaped defects
- **Issues closed vs opened in a given time**
- **Lead time vs cycle time**

Reasons:

1. Most metrics **focus on work completed** over a period of time, not on how it was delivered or if the customer gained any new business, affiliate, or could reduce cost of implementation, for example
2. The metrics are based on **limited data** provided by the supplier/provider. There is no analytics involved.
3. The KPIs are measure in an **ad-hoc** manner and often **without any benchmarks**.
4. The KPIs are not targeted towards outcomes, rather towards **output produced** by the supplier/provider

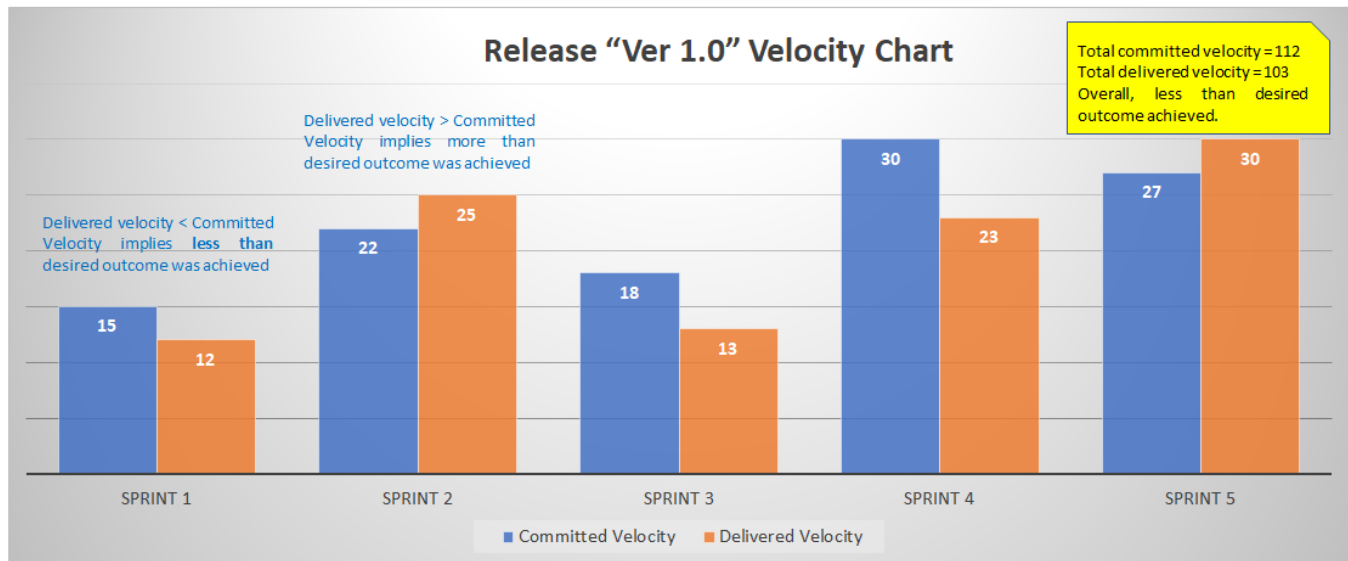
Q5. What KPIs are a good fit for OBC?

The same metrics as above and more are a good fit for OBC, if measured appropriately.

Metrics that take away the above concerns, are considered a good fit. Let us see how

- **Velocity** of stories that are linked to one or more outcomes. For details on how to attach an outcome to a Story, refer to point 10

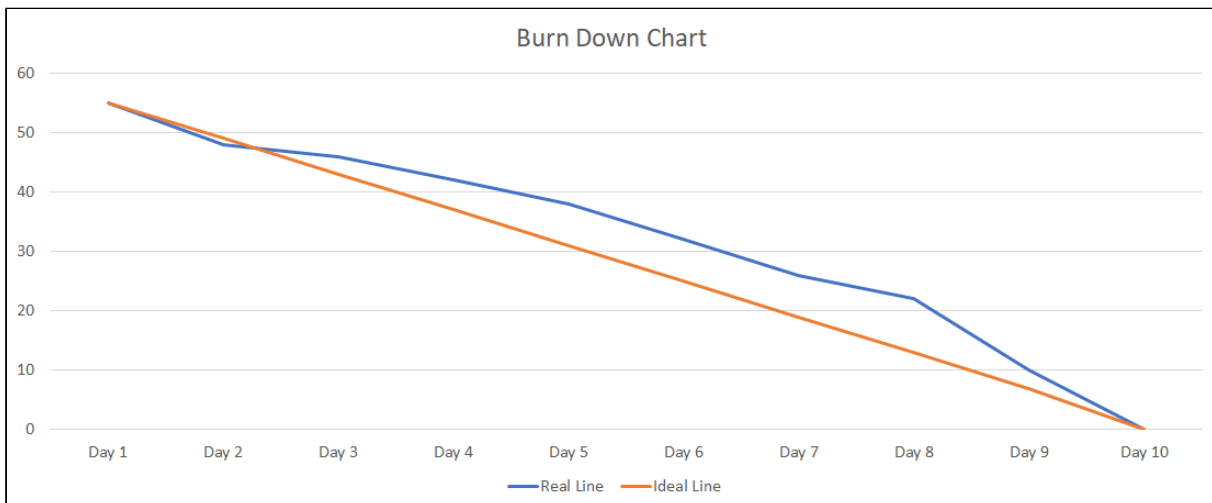
The Velocity chart below showcases how Outcome per Sprint and per Release can be tracked when it is tied to the Velocity



- **Escaped Defect Ratio**, which is ratio of escaped defects in the release to the velocity in the release

$$\text{Escaped Defect Ratio} = \frac{\text{Escaped Defects in Release "Ver1.0"}}{\text{Release "Ver1.0" velocity}}$$

- **Burn Down Chart**, which shows the number of hours left to burnt by the team on any specific day



Some very specific outcome based metrics or KPIs can be formulated under Outcome Based Contracts. These are attached to the respective outcomes.

1. Successful generation of the Patient Summary Report that is ready everyday before the daily meeting of the HCPs.
2. Login success report that shows how many login failures have been against the login attempts on any single day.
3. Successful receipt of issue logs at the customer and supplier ends of a secondary healthcare equipment that is not responding

Q6. Is time estimate and ongoing tracking still applicable and considered a good practice under OBC?

Time estimate and related tracking are very essential to building demonstrable KPIs.

For time estimation in OBC, please refer to Question 3.

Q7. How payment for work completed planned for in OBC?

One of the key reasons why OBC are popular is that they encourage payment as per achieved outcomes and not for achieved output.

One of the ways is to let the customer allocate some scope measured as per the outcome and some part of scope measured as per the output.

This way the payment work be planned for the supplier to create a balance.

Q8. Is OBC compatible to DevOps?

OBC is very much compatible with DevOps

Q9. How to work in Agile to support OBC? (methodology, etc)

By definition, Outcome Based Contracts follow all values and principles of Agile, so there is nothing more that needs to be done except encourage the customer and supplier to come together for an agreement and formulate the OBC

Q10. How to work with Jira to support OBC? (requirements hierarchy, etc)

By default Jira is well structure to capture all data to support OBC.

Here is one example, how the OBC can be captured in Jira.

Let us assume that the requirement is to build Daily Patient Summary Report for In-Patients everyday. One of the outcomes is to get this report ready for all patients before the meeting of HCPs.

1. The OBC is captured as an "Initiative"
2. The "Description" field is used to capture the agreement statement of the OBC.
3. The outcomes be captured under the field, "Acceptance Criteria"
4. Using appropriate User Story splitting pattern(s), multiple Stories may be created to accomplish the above outcome
5. One or more Stories can be combined to represent this outcome
6. The process as mentioned in point 3 can be followed to estimate the Stories
7. As the Stories get Done, the outcome of each is verified to ensure that the OBC is being followed

Key take-aways:

1. Outcome Based Contracts is for shared accountability of the customer and the service provider
2. Outcome Based Contracts focus on MVP

Reference::

Image 1: *Norton, Doc. Escape Velocity: Better Metrics for Agile Teams (p. 60). OnBelay Consulting, LLC. Kindle Edition.*"

what we've done is create three or four stories that cannot be independently delivered. They're all connected. They are one story.

Instead, focus on small stories that are thin vertical slices through the technology stack. Now that's easy enough to say and it sounds right, but it is not necessarily that easy to execute on. Sometimes, thin vertical slices look like waste. They cause us to build something less than what we'd envisioned. And we don't want to release anything less than perfect, do we?



Focus on small stories that are thin vertical slices through the technology stack.

I find it interesting how many teams I work with will simultaneously push for a Minimum Viable Product (MVP) and insist it is the right way to bring a piece of software to the market while arguing that writing a simpler version of the code is a waste of time since we already know what we need. An MVP is about learning: learning if what you *thought* the customer wanted was, in fact, what they wanted. An MVP is about getting something out there to validate your hypothesis and to learn. In that regard, thin vertical slices are merely MVPs taken to the next level.