WHITE PAPER

Implementing Agile Methodologies in Outsourcing Engagement Models and Measuring Success

VICERT

1. Introduction

Measuring the success and impact of development is a complex process, and figuring out the proper KPIs and searching for how to be better in each iteration is not always straightforward. Moreover, Agile methodologies are usually not fully applicable in engagements where two (or more) companies share various roles on a mutual project.

The main risks, such as blurry success plans, scope uncertainty, lack of proper architecture design, and bad development practices, derive from different company cultures, different working methodologies and unclear business goals.

In this white paper you will learn:

- How to measure the success and impact of development
- How to define KPIs in outsourcing engagement models
- How to apply agile methodologies and implement engagement best practices

Milos Kowatschki

Engagement Manager at Vicert

2. Common pitfalls

Product Ownership

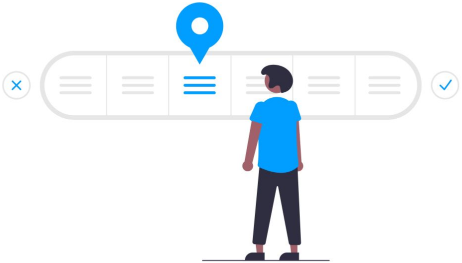
The first issue is how to handle product ownership. As we know, Product Owner is a vital role in agile methodology, as they are responsible for maintaining the team's product backlog, iteration planning, etc.
We learned from experience that having a Product Owner from the beginning of a project is critical to its success. The Product Owner does not necessarily have to be located in the product's company, however, they have to be fully aware of the product company's strategic goals and overall business strategy.

Lack of adequate architecture design

The second pitfall is a lack of adequate architectural design at the start of the product's development process. Ambiguous architecture and an imprecise tech stack could lead to project failure; sometimes it is better to spend development time in the development of PoC in order to verify that the right tech stack is chosen.
Currently, with such a vast technological landscape, it is easy to pick the wrong technology for a desired outcome.

Lack of Key Performance Indicators

The third common pitfall is lack of key performance indicators. Establishing KPIs for the development process and business success are not straightforward tasks. While these metrics may vary based on working methodology, type of team, and engagement model, it is paramount to do complementary work aligned with business goals in order to ensure project success. Important aspect of any process is successful measurement of its key performance indicators. It is hard to become more effective if you are unaware of baseline efficiency. Good practice is to provide extensive telemetry to a client concerning the development process. These KPIs may include team velocity, number of defects found by QA, merge request size, scope creep, backlog groom %, etc.



Blurry success plan

Apart from developmental efficiency, even more important is the impact a software product has on the company business. In our experience, many outsourcing companies are not attentive to the latter, even though we find the two to be inextricably linked.
Thus, before starting product development, you should create a clear product success plan in accordance with the client's business strategy.

Measuring development efficiency, engagement coordination, and business impact is vital for the development of a successful product. Lacking a definition of success and clear success plan usually leads to unsatisfactory product development which then leads to product business failure. When a piece of software is tightly coupled with the business achievements of the company, developing software that does not meet business demands leads to an inadequate ROI.

3. Working together with a client

As we mentioned earlier, important prerequisites for the success of a project are:

- Clear architecture design and tech stack
- Established product KPIs & product success plan
- Clear division of roles and responsibilities in compliance with Agile Scrum methodology

In our experience, the optimal way of working for dedicated teams that have a dual role -- working for the company and for our clients - is by following Agile Scrum methodology and executing development in two-week sprints.

Most commonly we participate in Sprint ceremonies with our clients, which enhances overall communication and collaboration. Since we are usually working in different time zones, we also adopted practices to communicate daily objectives and status reports via Slack channels at the beginning and end of each day. This is just a workaround for when we are not able to have daily stand-ups because of different time zones, or other spatial limitations.

On the following pages we will outline our methodology in practice -
please be aware that due to different time zones we are unable to fully adopt Scrum methodology.

Sprint planning

As previously mentioned, we perform a Sprint planning meeting together with our client. The main objective is to prioritize tasks and make sure every task has a story point.
We will use an excerpt from Atlassian blog to define story points:

Story points are units of measure for expressing an estimate of the overall effort required to fully implement a product backlog item or any other piece of work. Teams assign story points relative to work complexity, the amount of work, and risk or uncertainty. Values are assigned to more effectively break down work into smaller pieces, so they can address uncertainty. The idea is to have insight into the complexity of each task. Moreover, tracking how many story points a team is able to close in each iteration is essential for accurate forecasting.

Planning a Poker game

As we want to assign a number of story points for each task, the whole team has to somehow align on how many story points should be assigned for each given task. The way we do this is by playing Planning Poker, which is a consensus-based technique where each team member, at the same time, types - usually in the conference chat - how many story points they would appoint to a specific task.

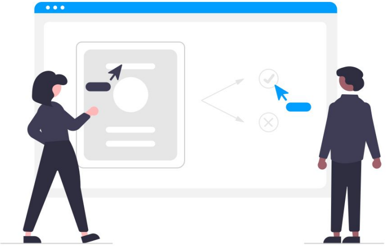
(There are also Planning Poker add-ons available for most collaboration tools).
Regarding the number of points, we have decided on fibonacci numbers from 1 to 8 - if a task has 8 or more story points, we need to break it into smaller pieces. Consequently, in one sprint, we don't have a task that has more than 8 story points.
Some companies use different practices where one story point is reflected as one hour of work. Consequently, the total number of story points is the number of hours necessary to complete a task.

Sprint review

The sprint review is used as a showcase for our client. We always try to have a demo available for a client so we can interactively show a project’s progress.

Sprint retrospective

During the sprint review meeting, we review what has been done during the sprint, with a focus on addressing issues and concerns we had. While addressing issues, we are looking for ways to resolve them in the next sprint, which consequently optimizes the development process.



Steering committee meeting

This meeting is not a ceremony which is part of the Scrum methodology but rather a stakeholder meeting in which we do a “retrospective” of our engagement. Such meetings are either performed after each sprint or at least once a month. The idea is to address risks, issues, and concerns during the engagement process. During the meeting, we always ask our clients to grade our engagement from 1 to 5. This allows clients to explain any grade we get that is under a 5, and creates a space for us to discuss issues, no matter how small. We find value in the fact that what may seem ‘insignificant’ at a given moment, could potentially become significant later on. In our experience, such methodology has proven to be a good health-check for client engagement.

Review your engagement frequently

In order to get processes better over time, we first need to establish them and measure their impact. In order to do this your processes should include:

Monthly review

Usually prepared on the last day of the month, this review should include metrics obtained from business and delivery presented in simple graphs/tables. The goal is to have a visualisation of the roadmap execution and relevant KPIs that would reflect code quality, velocity, status of the backlog, team dynamics, and client collaboration.

Quarterly evaluation report

For longer projects, a good practice is to prepare a quarterly evaluation report which would aggregate data from monthly reports along with business KPIs and the customer satisfaction score. The report should also address issues and concerns that arose throughout the project's execution. The goal is to perform an assessment of project performance and execution and present valuable insights to the client.

Project summary

This artifact is made on important milestones such as the launch of the MVP of the product, or the end of the engagement. The project summary should contain relevant data from quarterly and monthly reports along with a retrospective of the project execution. The goal is to show areas of improvement over time along with obstacles that have been overcome.

Implement best practices in delivery

Implement Continuous Integration

Continuous integration is a software development practice in which developers regularly merge their code into a centralized repository which triggers automated build and test processes. The goal is to improve software quality. By automating building and testing processes, commits are smaller, code is committed in regular intervals, and artifacts are being tested automatically.

Implement Continuous Delivery

Continuous delivery is a software development practice in which code changes are automatically built, tested, and prepared for the release. Usually, applications are running in different environments and in different configurations. Thus, by deploying all code changes to testing and production environments, software developers always have deployment-ready features that are deployed through a standardized process.

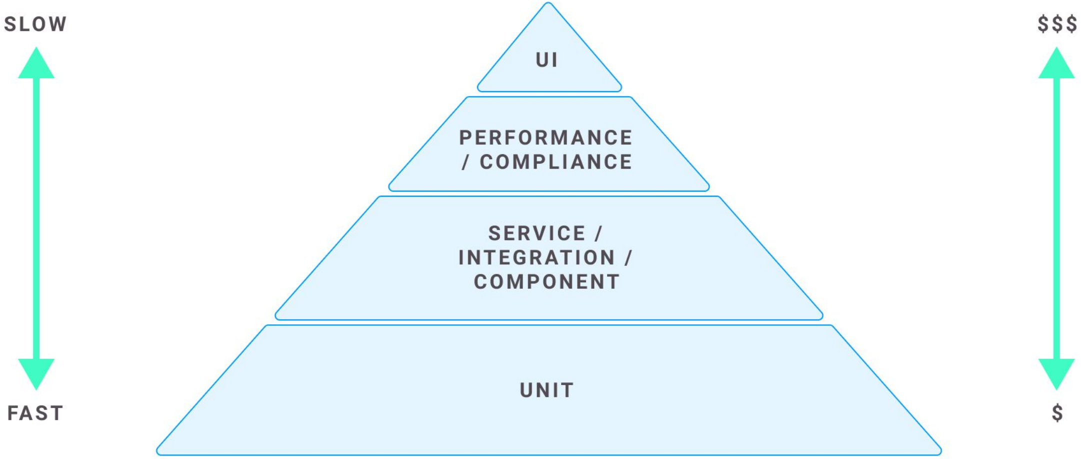
Some benefits are higher developer productivity, improved code quality, and more frequent updates.

By adopting CI/CD models, we are looking to have structured well-defined processes which rely on various testings, such as integration testing, component testing, system testing, performance testing, compliance testing, and user acceptance testing, to obtain better quality of product development.

As seen below, unit tests are at the bottom of the pyramid; they are the fastest and cheapest to run. Consequently, unit tests should cover approximately 70% of your testing efforts.

The most expensive and slowest are UI tests as they require production quality environments to run.

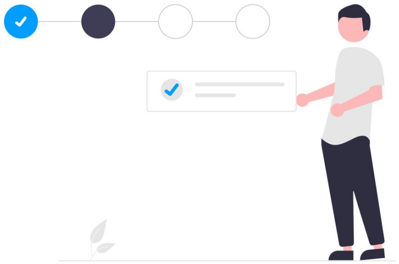
CI/CD TESTING PYRAMID



REF: BASED ON SUCCEEDING WITH AGILE BY MIKE COHN

In order to find the right balance, you need to assess each project independently and take into consideration your resources and budget.

It is crucial to communicate to the client the importance of test-driven development and acquiring best practices in development. Usually, clients tend to insist on skipping these steps because of budget constraints. However, the price gets much higher for product improvement and maintenance down the road.



To wrap up:

- Use version control and establish a proper tracking/ticketing system. Moreover, have peers review changes before applying them
- Put teams of no more than 12 self-sustaining members and have them commit small changes in frequent cycles
- Make sure that unit tests cover at least 90% of the code but try to go for 100% and keep them up to date
- Establish role-based security controls
- Keep track of metrics such as: number of builds, number of deployments, build time, etc.

4. Conclusion

There are no methodologies that work for every engagement model. The key point is that the whole team needs to have an understanding of their role in enabling business success.

This methodological point goes hand in hand with understanding responsibility, ownership, decisions, and who has the authority to make them. The end goal is that the whole team supports each other in the fulfillment of a business outcome.

In order to find the proper engagement model, you would probably need to go through a series of trials and errors. However, that does not mean that we are unaware of best practices.

By utilizing best practices and performing a series of trials and errors in your engagements, that iteration and amelioration will naturally bring you to an optimized model.

SOFTWARE CURES THE WORLD

CONTACT

www.vicert.com
info@vicert.com
@Vicert_Inc

SAN FRANCISCO

1355 Market Street, Suite 488
San Francisco, CA 94103, USA
+1.415.4957700