**Ideation Phase**

**Define the Problem Statements**

|  |  |
| --- | --- |
| Date | 06 May 2023 |
| Team ID | NM2023TMID18687 |
| Project Name | Assessing the safety of municipal drinking water |
| Maximum Marks | 2 Marks |

**Customer Problem Statement Template:**

In your time as a product manager, it is likely that you and your team will face many different customer pain points, needs, and opportunities. Problems (and the reasons why they occur) always seem to be never ending — whether it’s customers complaining about your product’s poor user experience, its high cost, or other issues that seem innocuous to you but serious to your customers.

There are a number of different problems for your team to solve, and these problems can also be poorly defined from a customer perspective. This implies that it’s difficult to figure out what you and your team need to do to remedy the vague or little known pain point.

As a product manager, it is your responsibility to help define the problem, to scope and set boundaries over it, and to point to a definition of success for resolving that problem. This helps your team understand the issues that your customers are facing, ideate potential solutions, and make necessary trade-offs.

The description, definition, and scope of the problem can be succinctly described as part of a problem statement. In this article, we will walk through what a problem statement in product management is, the advantages of having a well-defined problem statement, key frameworks to use when defining your problem statement, and elements of an effective problem statement.

## Table of contents

* [What is a problem statement?](https://blog.logrocket.com/product-management/problem-statement-template-examples/#what-is-a-problem-statement)
* [The advantages of having a well-defined problem statement](https://blog.logrocket.com/product-management/problem-statement-template-examples/#the-advantages-of-having-a-well-defined-problem-statement)
  + [Helps see the problem from the customer’s perspective](https://blog.logrocket.com/product-management/problem-statement-template-examples/#helps-see-the-problem-from-the-customers-perspective)
  + [Creates an understanding of the job that the customer is trying to do with your product](https://blog.logrocket.com/product-management/problem-statement-template-examples/#creates-an-understanding-of-the-job-that-the-customer-is-trying-to-do-with-your-product)
  + [Keeps the team focused on delivering a real solution to solve a real problem](https://blog.logrocket.com/product-management/problem-statement-template-examples/#keeps-the-team-focused-on-delivering-a-real-solution-to-solve-a-real-problem)
* [Frameworks to use when defining a problem statement](https://blog.logrocket.com/product-management/problem-statement-template-examples/#frameworks-to-use-when-defining-a-problem-statement)
  + [Writing from a user persona perspective](https://blog.logrocket.com/product-management/problem-statement-template-examples/#writing-from-a-user-persona-perspective)
  + [The Jobs-to-be-Done (JTBD) framework](https://blog.logrocket.com/product-management/problem-statement-template-examples/#the-jobs-to-be-done-jtbd-framework)
* [Problem statement template](https://blog.logrocket.com/product-management/problem-statement-template-examples/#template-for-an-effective-problem-statement)
* [Examples of realistic problem statements](https://blog.logrocket.com/product-management/problem-statement-template-examples/#examples-of-realistic-problem-statements)

## What is a problem statement?

As mentioned in the name, a problem statement is a written statement about the customer problem — usually expressed as a pain point, need, or opportunity — that you and your team are trying to resolve. You can do this by either coming up with a technical solution ([like an initiative that becomes epics](https://blog.logrocket.com/product-management/what-is-an-epic-in-agile-guide-examples/) and user stories) or consulting about it with another function of the organization. For example, if the pain point relates to pricing, it’s best to consult with your revenue and account management team.

Although it sounds simple in theory, it can be a difficult and frustrating exercise in practice. Firstly, problems can be ill-defined by the customer in the first place — meaning that it doesn’t necessarily have the shape and structure to enable your team to find a well-scoped solution. Furthermore, there may be a number of different hidden problems masking the “real reason” the issue occurred in the first place.

As such, time needs to be invested to figure out if the problem presented by the customer at first instance is the real and only problem, or if there are layers underneath that need to be explored to determine if there is a deeper, systemic issue instead.

It helps to have a well-structured, evidence-based problem statement that allows your team to dial into the actual problem. A focused solution can then be implemented to resolve the real or foundational customer need, pain point, or opportunity presented by the problem statement.

## The advantages of having a well-defined problem statement

There are a number of advantages to having a well-defined problem statement. We’ll go over them in detail below.

### Helps see the problem from the customer’s perspective

Problem statements are usually written from the customer’s point of view. That is, it usually considers the types of problems that the customer faces because of the “life role” they are currently using your product for. Whether your product is a web or mobile application, whether it’s for consumer or business consumption, or whether it’s a SaaS product, chances are that your customer is facing the problem with your product based on what they are at that moment.

For example, a customer of a social media application will have different problems, pain points, needs, or opportunities compared to a customer of a stock trading platform application. They are trying to do different things on each individual app, and due to this, they will run into different problems that only they will experience as the bespoke customers of your application.

## [Over 200k developers and product managers use LogRocket to create better digital experiences](https://lp.logrocket.com/blg/pm-demo-signup)

**[[](https://lp.logrocket.com/blg/pm-demo-signup)Learn more →](https://lp.logrocket.com/blg/pm-demo-signup)**

### Creates an understanding of the job that the customer is trying to do with your product

Further along to the first point, by seeing the role that your customer is playing at the moment of using your product, you also get the chance to understand the job that they are trying to complete by using your product.

A job in this sense is different from what you might think about traditionally — this is the thing that they are trying to complete at that point in time. The only way that they can complete the thing is by using the functions and features from your product.

By understanding the job that your customer is trying to achieve with the product, along with the “life role” they are playing when using your product, your team will have a unique understanding of the problems that they are facing and the reasons why those problems are serious. They are preventing the customer from getting the job done.

Say you are a social media user. A possible job that you want to get done is to post a picture on the internet for your friends to see. To do that, you need to have a function or feature on the app that allows you to choose a picture from your camera roll and possibly have a chance to edit or tweak the photo before posting it online. However, if the upload button is not working, you can’t post your pictures on the internet and, as a result, can’t get your job done.

### Keeps the team focused on delivering a real solution to solve a real problem

As we wrap up to the above two points, by understanding both the role and job that the customer is trying to achieve, your team focuses on the actual pain point and translates this focus into a viable solution.

This helps the team avoid the build trap, e.g., building functions and features for the sake of building. Instead, a well-written problem statement should help them really understand the “why” and “what” they are building, as well as the connection that the solution has to the pain point, issue, or opportunity.

## Frameworks to use when defining a problem statement

In structuring a proper problem statement, it can help to fall back on several tried and tested frameworks, methods, and theories.

The two frameworks that I employ most when writing a problem statement are:

* [Writing from a user persona perspective](https://blog.logrocket.com/product-management/problem-statement-template-examples/#writing-from-a-user-persona-perspective)
* [The Jobs-to-be-Done (JTBD) framework](https://blog.logrocket.com/product-management/problem-statement-template-examples/#the-jobs-to-be-done-jtbd-framework)

### Writing from a user persona perspective

A user persona is a fictional profile based on your real life user’s traits, which should be a [reflection of your product’s typical customer](https://blog.logrocket.com/product-management/buyer-personas-templates-and-examples/). By having a well-developed user persona, a product manager is capable of understanding the key traits, goals, and responsibilities of their typical customer. This enables them to translate that understanding into problem discovery and focus from a customer’s perspective.

In the context of developing a problem statement, a user persona is useful to assist you in understanding the exact job that they want to complete on your application or product. By understanding the job that they want to get done based on the goals and traits of their user persona, you will gain deeper insight into the real reasons why they are experiencing the problem and how best you can solve it.

### The Jobs-to-be-Done (JTBD) Framework

Based on Anthony Ulwick’s book What Customers Want, [the Jobs-to-be-Done (JTBD) framework](https://blog.logrocket.com/product-management/what-is-jobs-to-be-done-framework-examples/) stems from the idea that customers buy products and services to get the job done. In using your product, a customer will decide whether or not they will purchase or continue using your product based on how well it delivers on the outcomes that they are looking for, e.g. the job that they want to get done by using your product.

Using the JTBD framework together with a well crafted user persona provides you a holistic view of the customer, what they want to do with your product, why they want to do that particular job using your product, and the current problems preventing them from getting said job done using your product.

In this way, you help narrow your problem statement down to issues that, if resolved, will help with the resumption or increased frequency of the customer getting the job done using your product.

## Problem statement template

Using the frameworks above, a typical problem statement sounds something like this:

As a [USER];

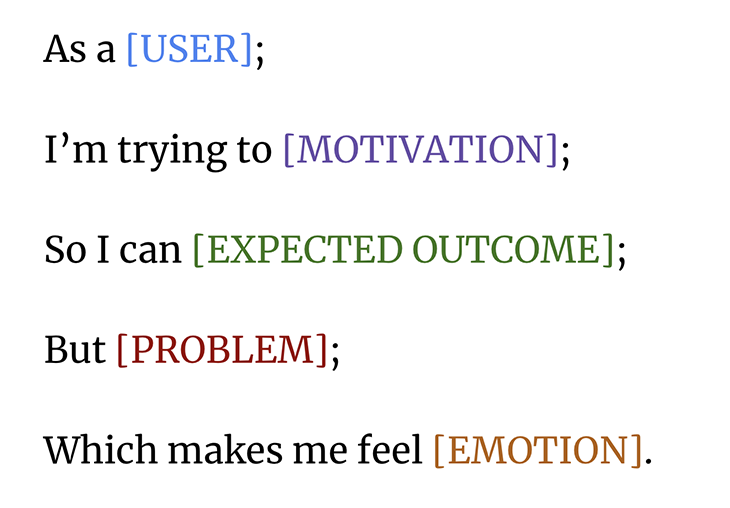
I’m trying to [MOTIVATION];

So I can [EXPECTED OUTCOME];

But [PROBLEM];

Which makes me feel [EMOTION].

You can use that as a template to write successful, actionable problem statements. You don’t need anything super fancy, as long as you hit on these points to get a holistic view of the problem:



The following is a breakdown of how we write this:

|  |  |
| --- | --- |
| **Element** | **Description** |
| **[USER]** | **This is based on your customers persona, e.g., a portrait or makeup of what your typical customer looks like.**  **This can be their job ( a manager, an engineer, etc.), their life stage (child, teenager, etc.), or any other trait that reflects who you are trying to sell your product to.** |
| **[MOTIVATION]** | **This will be informed by the job the user is trying to accomplish via the JTBD framework. It’s usually something that your user is trying to do by using your product but can’t because of the problem in your product.** |
| **[EXPECTED OUTCOME]** | **The outcome is the exact job that they are trying to accomplish, which comes from your JTBD framework.**  **For example, if you were an engineer and were using a laptop, your expected outcome would be that the laptop was performant enough to handle your workload. If the laptop is slow or crashes intermittently during the day, it is likely that this is blocking you from your expected outcome and, hence, is a problem or pain point to solve.** |
| **[PROBLEM]** | **It is as it says — this is the exact nature of the problem.**  **The problem here should ideally be simple to describe and doesn’t require much context, as most of that context should’ve been completed in the user, motivation, and expected outcome phases of the problem statement.** |
| **[EMOTION]** | **Finally, the emotion is the mental state of the customer that is affected by not being able to achieve the expected outcome.**  **Although this might sound trivial, the emotional state of a customer can sometimes subtly influence how your customer perceives your product, whether positively or negatively, and dictates how often your customer can convince themselves to use your product to get the job done.**  **It is important to consider their frustration, anger, or disappointment. This not only informs you about the severity of the problem, but also, by bringing emotion into the equation, your team can feel exactly what the customer is feeling, motivating them to provide a solution that solves their problem.** |

## Examples of realistic problem statements

Going off of the previous section where we looked at a problem statement template, let’s now review some examples:

|  |  |
| --- | --- |
| **Scenario** | **Problem statement** |
| **A TikTok user is unable to save their videos into a personal list of favorites** | **As a [TikTok user];**  **I’m trying to [save a video I just viewed];**  **So I can [watch the video in my own time or share it with friends & family];**  **But [I am not provided the option to save];**  **Which makes me feel [frustrated].** |
| **An iPhone user is unable to take good pictures at night** | **As an [iPhone user];**  **I’m trying to [take a picture at night];**  **So I can [capture a shot that only looks good in dim to no lighting];**  **But [the phone does not take good night shots];**  **Which makes me feel [sad].** |
| **A popular application you use does not have a native Slack integration** | **As a [Slack user];**  **I’m trying to [include Slack notifications when certain actions are triggered in the application];**  **So I can [be alerted when something wrong happens];**  **But [I am not able to connect my application to Slack];**  **Which makes me feel [stressed].** |

## Conclusion

Follow the above tips and you’ll be writing expertly crafted and well defined problem statements in no time. Thanks!

Featured image source: *[IconScout](https://iconscout.com/icon/file-3250251)*

## [LogRocket](https://lp.logrocket.com/blg/pm-demo-signup) generates product insights that lead to meaningful action

[LogRocket](https://lp.logrocket.com/blg/pm-demo-signup) identifies friction points in the user experience so you can make informed decisions about product and design changes that must happen to hit your goals.