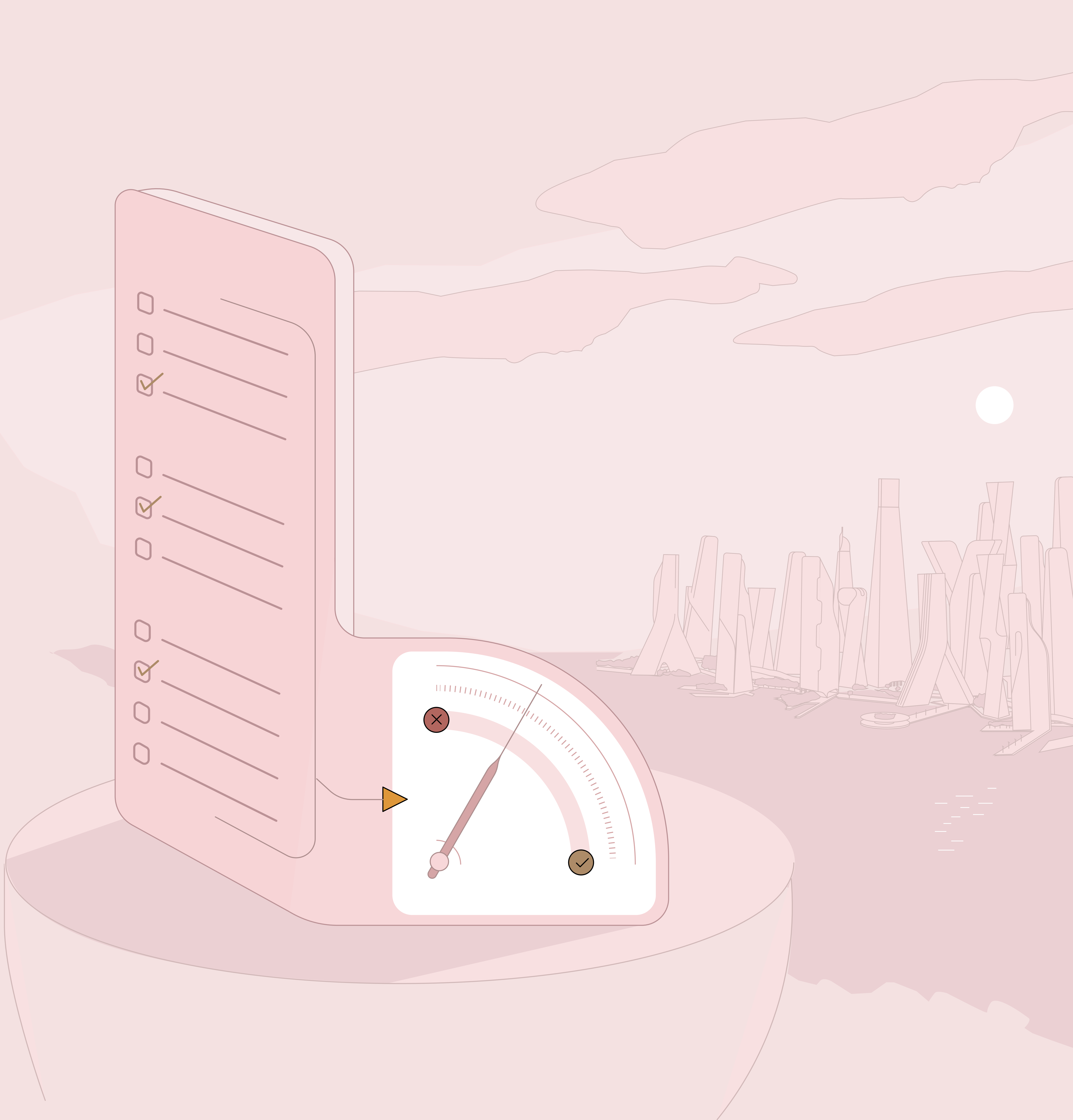
tech health evaluation sheet

asess the technical health of your digital product



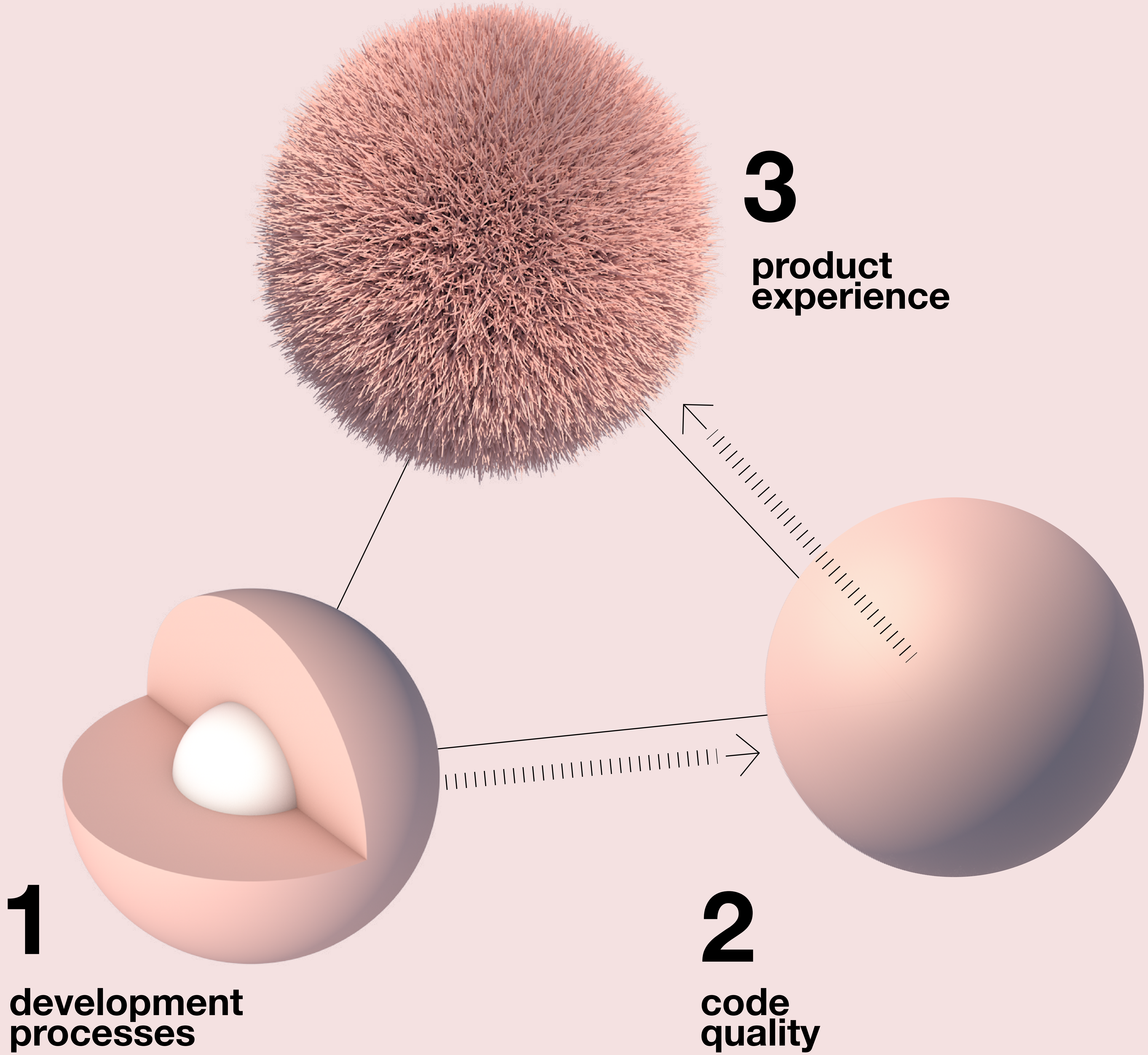
SUPER CHARGE

having a healthy tech set-up is not an advantage anymore — it’s the bare essential

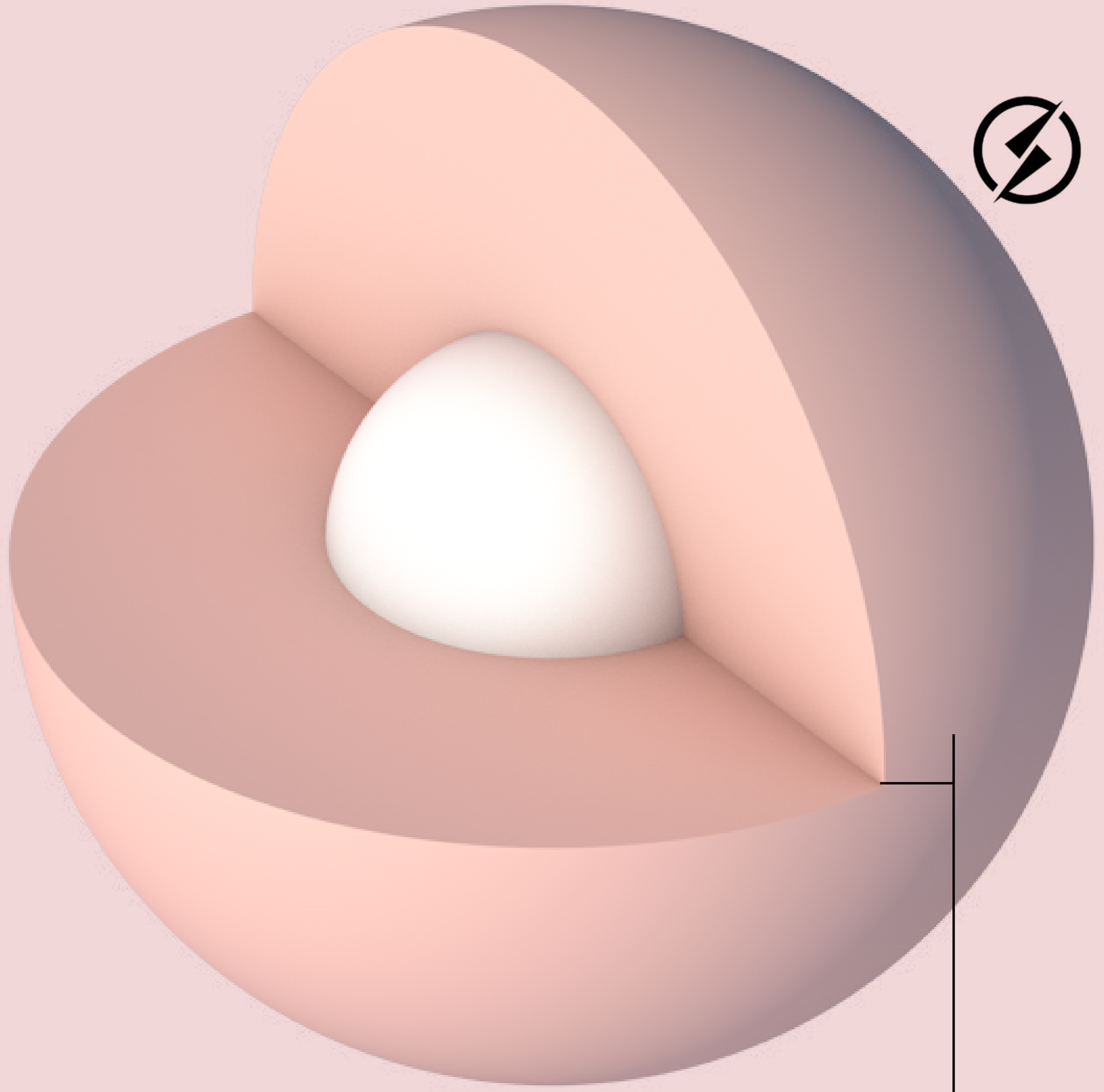
is your product built on rock-solid technological foundations?

fill out our self assessment sheet to figure out if your digital solution suffers from any symptoms of bad technology decisions.

we’ll lead you through 3 areas where these symptoms usually reveal themselves: development processes, code quality and product experience.



1 developement processes



1.1 let’s assess how reliable is your release schedule.

can you release new application versions according to schedule?

A new release means that the development team needs to upload a new version into the app store (in case of a mobile app) or to your webserver’s production environment.

0pt never, we are 1 month late every time

1pt we often have a few days or a week of delay

2pt we are punctual like a Swiss watch - on time, every time

1.2 let’s assess how reliable your Quality Assurance processes are.

how good is your QA at catching bugs before your users?

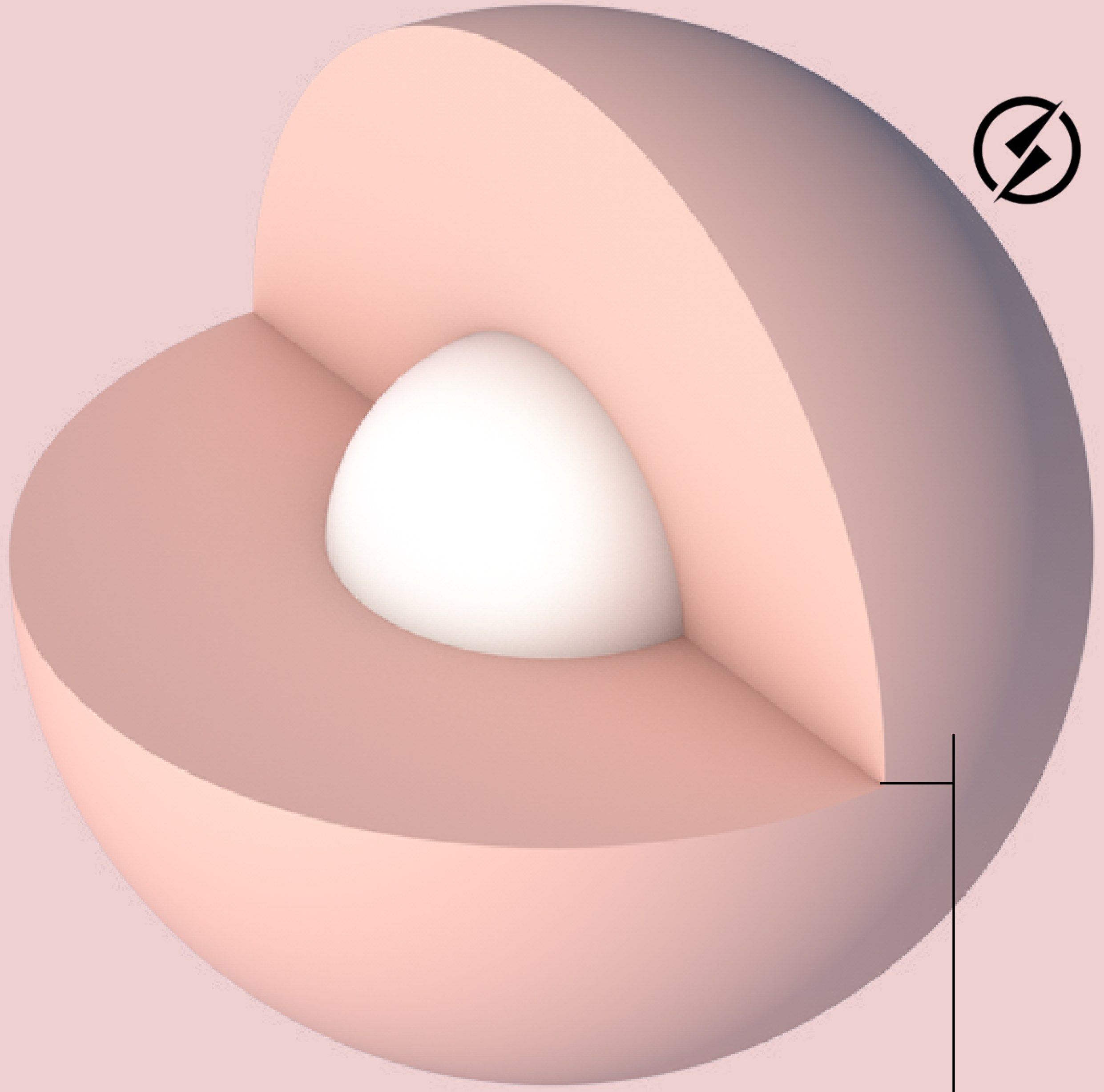
Strong QA means that bugs very rarely make it to the customers. If they do, either your QA processes have issues or the QA-dev communication is broken.

0pt our users frequently report serious bugs

1pt our user infrequently report serious bugs, and sometimes minor ones (cosmetic or annoying things)

2pt almost have no bugs in our new releases

1 development processes



1.3 let's assess how efficient your Quality Assurance processes are.

are there automated tests that run before every application release?

While manual testing is always advised, automated testing can introduce unparalleled efficiency and predictability to your QA processes. Simply ask your dev team about the coverage of automated tests.

0pt	we don't have any automated tests
1pt	we do have automated tests, but they don't cover the whole application
2pt	the robots are busy testing all moving parts before every release

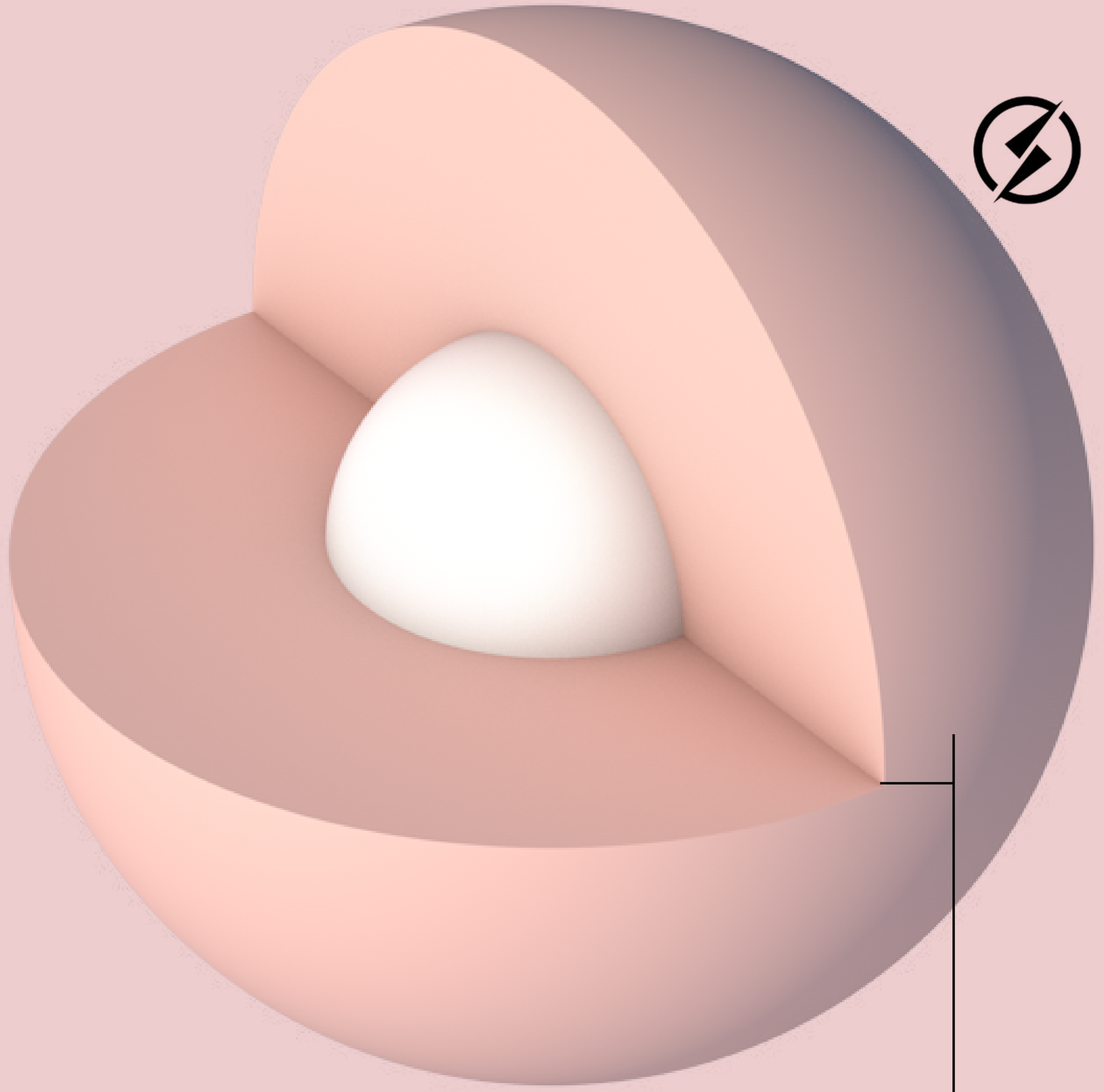
1.4 let's assess the information flow between business and development.

how easy is it for you to get information about the state of readiness and expected delivery date of a feature?

While development might encounter unforeseen blocks, keeping decision makers well informed is essential to running an effective business operation.

0pt	it usually seems impossible to get a clear answer to these questions
1pt	I am informed, but it takes extra effort on my side, and the information I get is not always useful
2pt	I am always well informed and can use that information to make business decisions

1 development processes



1.5 let’s assess how well documented the application is.

how easy is it to onboard new developers?

Documentation is essential for growing teams as it makes learning the ropes much easier for new team members, and explaining all the details of the code and architecture doesn't take away time from senior team members.

0pt	it is resource intensive, because there is no documentation and everything has to be learnt from the team members
1pt	there is some documentation that helps with onboarding, but it only covers part of the application, or hasn’t been updated for a while
2pt	onboarding is a breeze as everything is beautifully documented

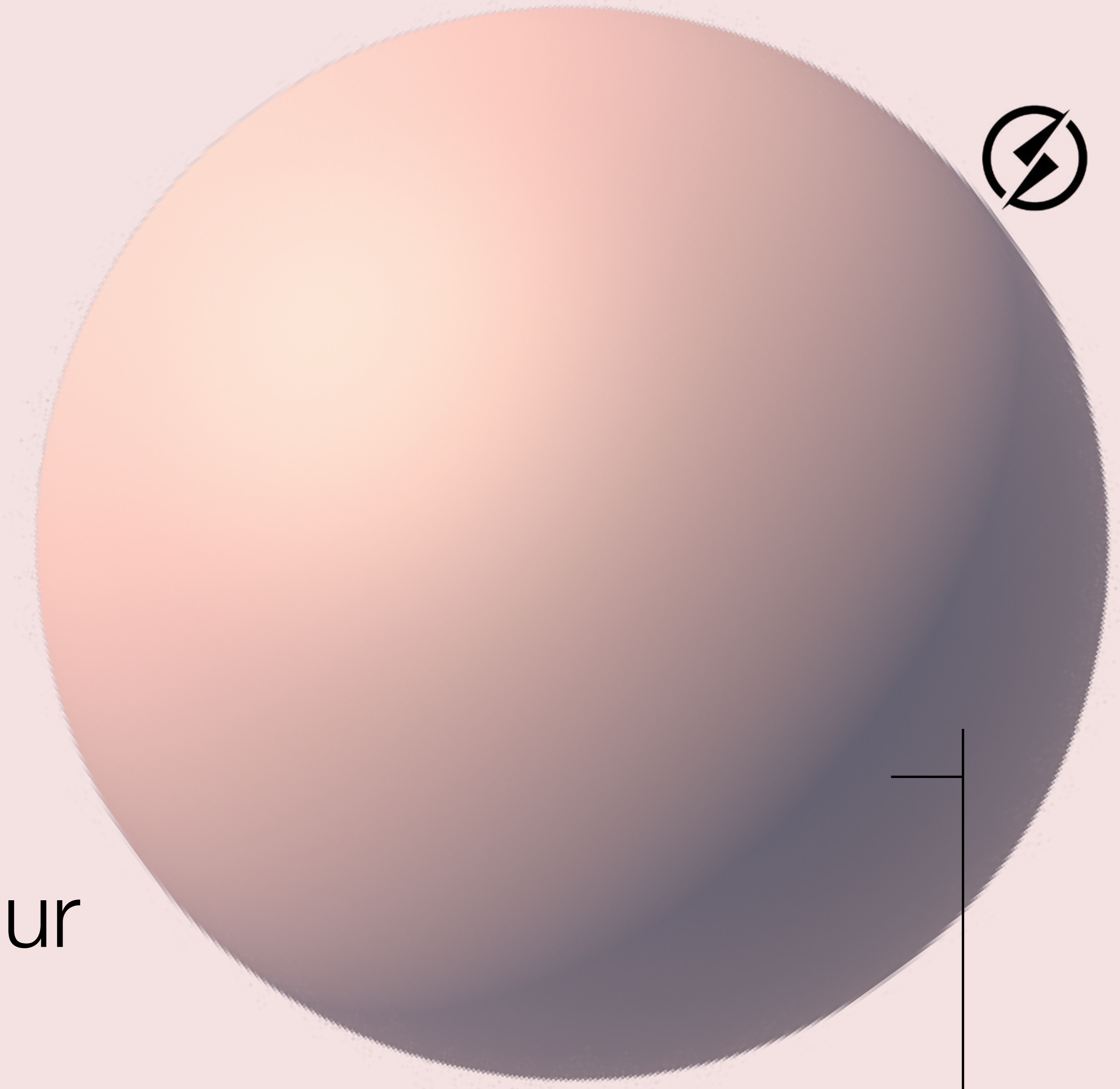
1.6 let’s assess the state of the test environment.

is there a test environment where you can test every feature before it goes live?

Having the right environment setup is very important for the QA and for various business side stakeholders to make sure every feature hits the market as intended.

0pt	there is no such environment, or it is not really accessible
1pt	there is a test environment but some features can only be tested in the live environment
2pt	there is a test environment where I can test any feature before release

2 code quality



2.1 let’s assess how simple it is to add new elements to your codebase.

how fast is it to introduce new features to the application?

0pt — **takes a lot of time on every occasion, even if the new feature’s complexity doesn’t seem to justify that**

1pt — **the speed seems okay, but the developers often run into unforeseen complications**

2pt — **it is relatively fast**

2.2 let’s assess how safe it is to add new elements to your codebase.

does the app “break” when new features are added or features are modified?

By “breaking”, we mean unexpected errors and bugs surfacing - not necessarily within the new feature. You get the feeling that you pull one cord and something else breaks at the other side of the code…

0pt — **it happens all the time, new features & modifications constantly break the application, and the team needs to fix errors.**

1pt — **new features & modifications sometimes break the application and bring unexpected errors to light**

2pt — **new features & modifications are added to the application without causing unexpected errors and crashes**

2 code quality

2.3 let's assess how the development team feels about the codebase.

Working in a "spaghetti-code" is never fun, and can be severely detrimental to employee experience in the development team, hurting retention and increasing burnout.

how many complaints do you hear from the developers regarding code quality?

0pt	I feel that I hear nothing else…
1pt	this complaint sometimes comes up in discussions with the dev team
2pt	I hear almost no complaints about the code quality from devs

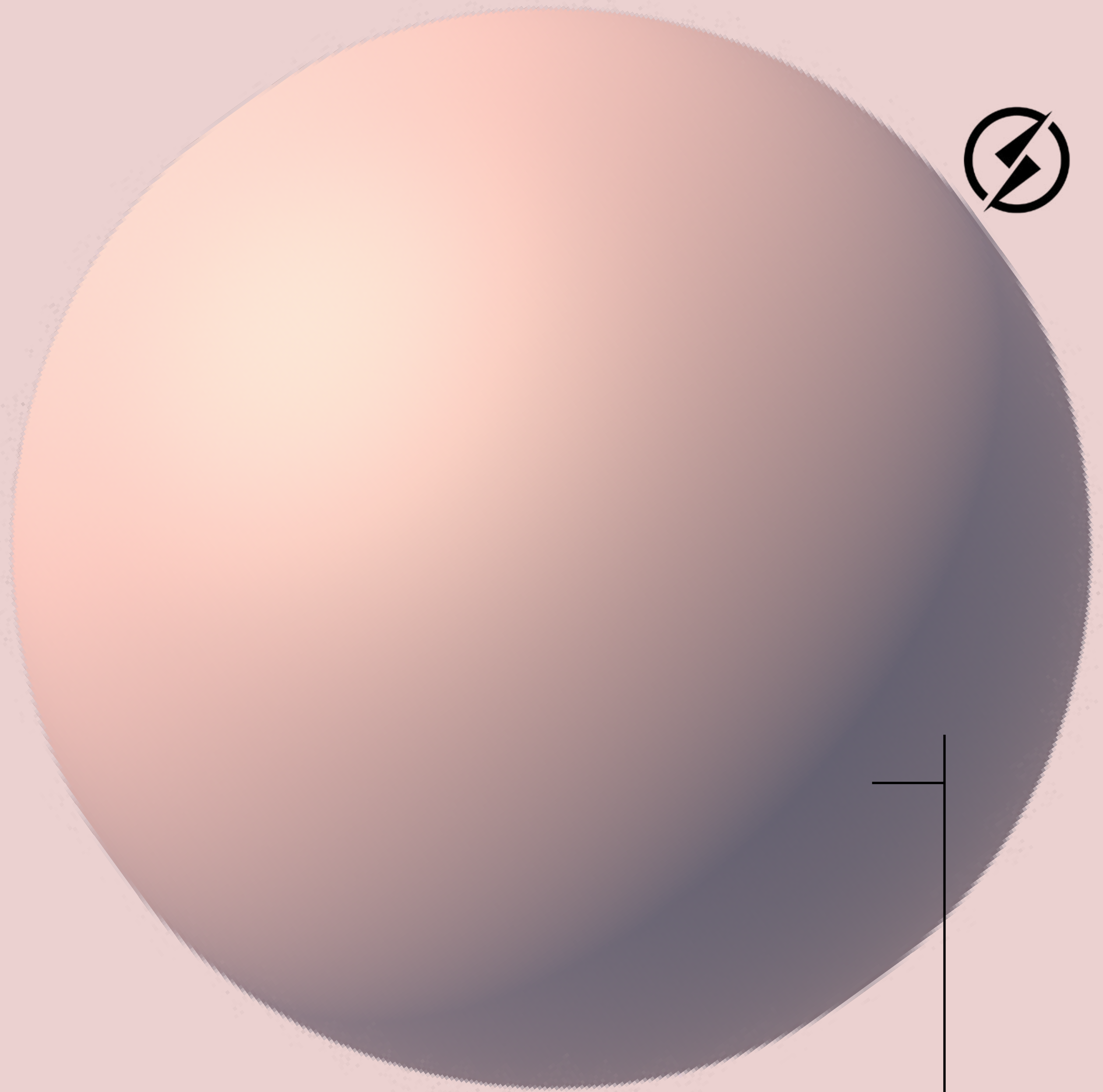
2.4 let's assess how smartly your app handles content management.

if you want to change contents in the app (items, articles, rich media, etc.), do you need to issue a new release?

A new release means that the development team needs to upload a new version into the app store (in case of a mobile app) or to your webserver's production environment.

0pt	we need to issue a new release for any content change
1pt	we can change some of the content elements without a new release, but not all of them
2pt	we can change every content element without issuing a new release

2 code quality



2.5 let’s asses how much your application takes advantage of industry standard, 3rd party components.

does your app use reliable third-party software components, also known as libraries?

Third-party libraries guarantee that your solution is built on a reliable code base, which has been tested and approved by the developer community.

Ask your development team for a list of libraries used if you have problem answering this question.

Disclaimer: There are certain cases when deliberately avoiding 3rd party libraries is the right choice. If this is the case for you select 2 pt.

0pt **no, every line of code has been written by our own team and there is no sign of trying to leverage 3rd party libs**

1pt **some 3rd party libs are used, but the tech teams approach is not “check if a reliable open source solution exists - only write it, if doesn’t”**

2pt **the tech team leverages industry standard 3rd party libs whenever it is a viable option**

2.6 let’s assess how smartly your app takes advantage of modularity.

if developers update one element in one part of the app, is the same element updated all across?

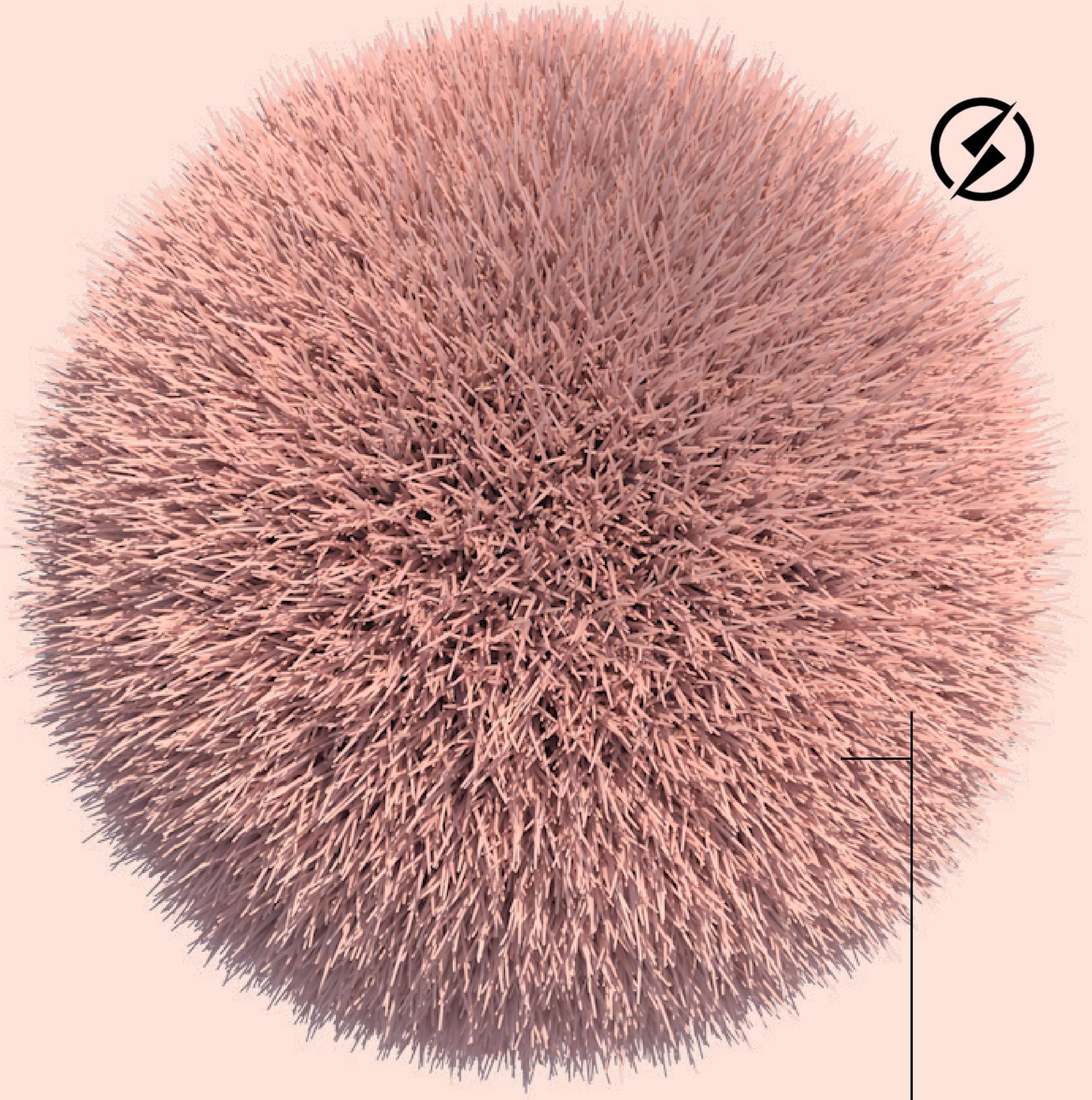
Reusing modules is a good practice because it decreases development time and keeps things manageable. For example, think of a green button - if you wanted to change all your green buttons to blue, do your developers have to go through them one by one?

0pt **every element needs to be updated separately, even if it’s the same module**

1pt **there are some important elements that can be updated at once, but not everything**

2pt **it is a marvel of efficiency, everything that looks the same updates from one place**

3 product experience



3.1 let's assess how reliable your app feels to your customers.

are there any critical errors making it impossible for users to access the app or its functions?

Critical errors are the meanest of any errors - they stop users from doing what they intended to do in the software. Think of the app suddenly quitting at some point, a page not loading, or a process getting stuck forever.

These are the ones that really erode customer experience and trust in your service, and can be very uncomfortable for the user. The more important your app is for the user (kudos for that!), the more devastating critical errors feel (ouch).

0pt	**we often get reports about critical errors**
1pt	**we occasionally get reports about critical errors**
2pt	**I don't even remember the last time a critical error was reported...**

3.2 let's assess how well-built your app feels.

are there any errors that make your app look "shabby"?

Think of misaligned elements (text, boxes, buttons), elements overlapping each other, or elements that look bad on certain screen sizes. Anything which definitely doesn't look like the original design.

0pt	**the app is full of such issues, I can spot one on almost every screen.**
1pt	**the app mostly looks like the designs, but I can easily spot some issues**
2pt	**nope, it is picture perfect on every device it was intended to run on**

3 product experience

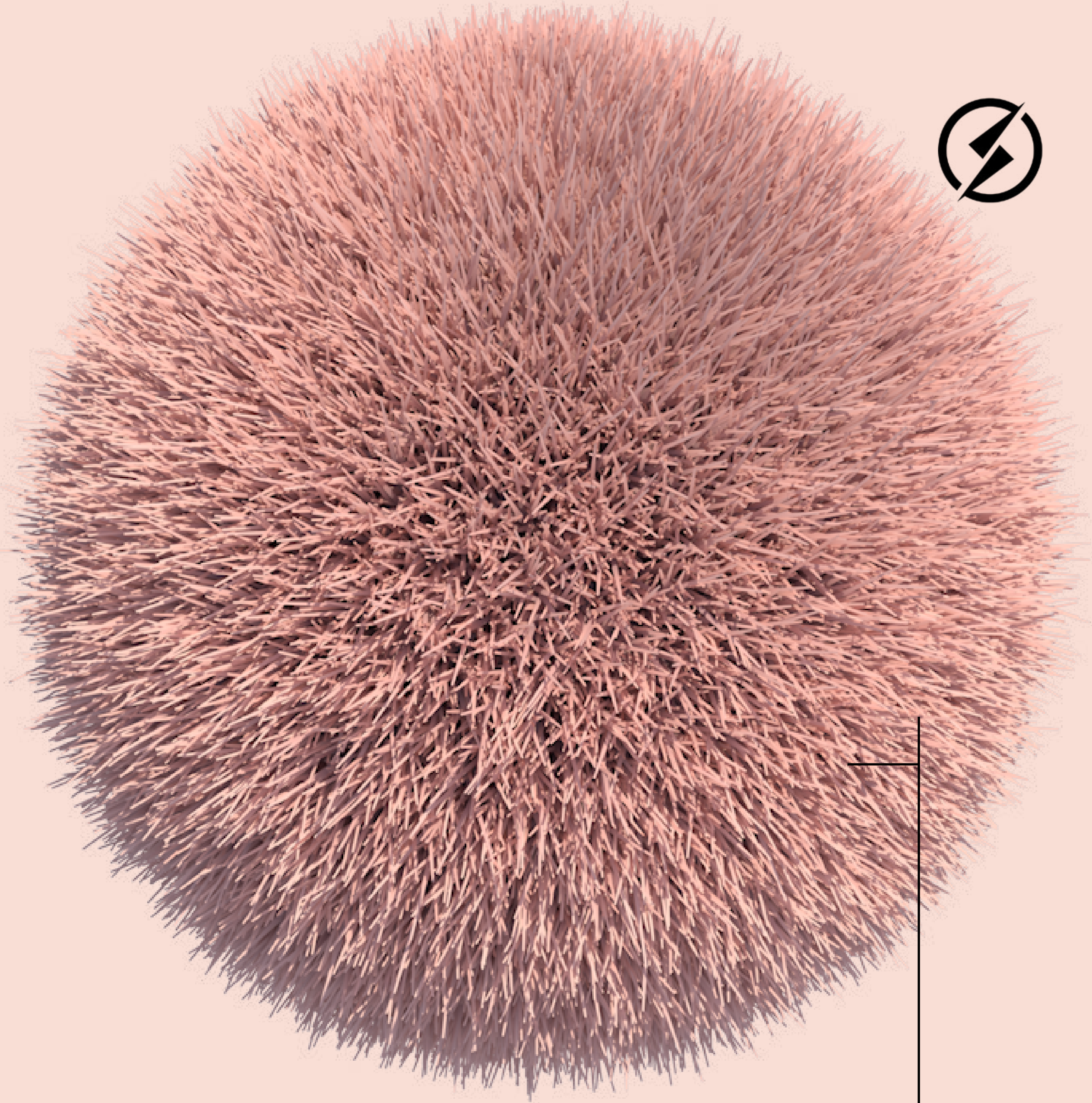
3.3 let's assess how much your users need to wait.

how performant is the app when it comes to loading times?

Whenever the application seems to wait for data and the user can't continue their activity in it - well that's loading time, and everyone hates it. Depending on the context and how often the user sees it, even a few seconds of loading can severely decrease the user experience.

- 0pt **there is a lot of loading on frequently visited screens/pages. Even I'm bored with it when I test the app.**
- 1pt **there are some screens/pages that take time to load, but they are less often visited**
- 2pt **it is blazing fast**

3.4 let's assess how well your application handles workload.



are there issues in peak periods, when lots of users use the application?

If the experience offered by your application decreases over a certain number of simultaneously active users we talk about performance issues. It can mean services becoming unreliable (login not working, etc.), or certain things taking much longer time to load (or don't load at all).

- 0pt **the application's performance and/or reliability clearly decreases in peak periods, and we don't need too many active users to call it a peak...**
- 1pt **the application's performance and/or reliability clearly decreases in peak periods, but we rarely reach that number of simultaneously active users**
- 2pt **at this point, the application's performance is not visibly affected by the number of simultaneously active users**

3 product experience

3.5 let's assess how swift the interface feels.

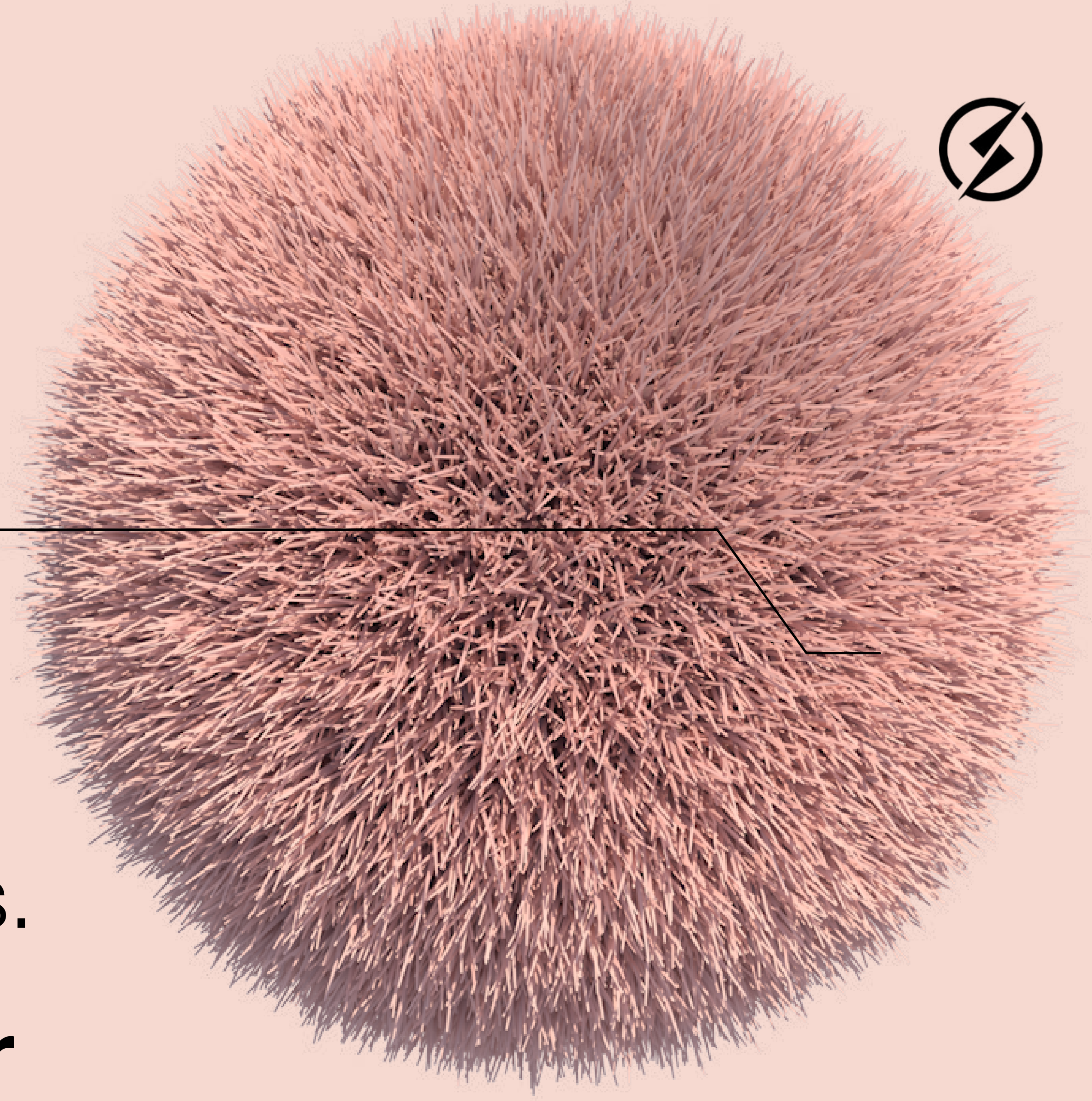
how smooth are animations and screen transitions in the application?

Smooth user experience requires… well, smooth motions. It is important to test this on a device with average performance - average can mean very different things based on your customer base. Look out for interactive diagrams, carousels, scrolling, transitions between different screens of the application. Technically speaking, we usually consider 60 FPS (frame-per-second) to be fully smooth for the human eye, but you can probably judge it without any measurements.

0pt a lot of moving elements seem to be stuttering, even some basic interactions (like scrolling) feel clunky and/or new screens build up with clear stutter

1pt there are some stutters and clunky animations in the application

2pt it's all smooth like butter



3.6 let's assess how reliable your user data management is.

do your users ever experience data loss or are they sometimes unable to access their data?

Think of things like files, favorites, pinned items, or progress made. Entering an app and not seeing your data is a major source of frustration.

0pt every time the app is opened, it's like an untouched territory. There's no sign of anyone ever using itit; user data is never saved.

1pt there are one or two minor data types that are not saved properly, but these functions are rarely used by the user.

2pt everything is in perfect sync like a philharmonic orchestra

how healthy is your tech foundation?

how do i calculate my score?

to get an overview of your digital solution's tech health, add up your scores belonging to each of your answers.

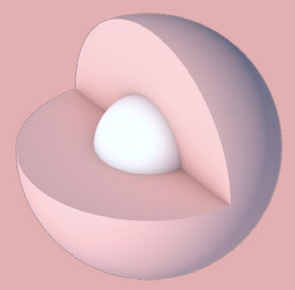
identifying your weak points

Identifying the blind spots of your software's digital foundation is your step 0 on your journey towards reaching your business ambitions.

1. **Sum up your points for each subcategory:** To learn about where to concentrate your efforts and understand where your digital product lags behind, add up your subscores from the product expercience, code health and development processes sections.
2. **The areas under 6 pts are your weak points:** Even if your overall scores look good, just one weak pillar can hamstring your performance. Do you have any?

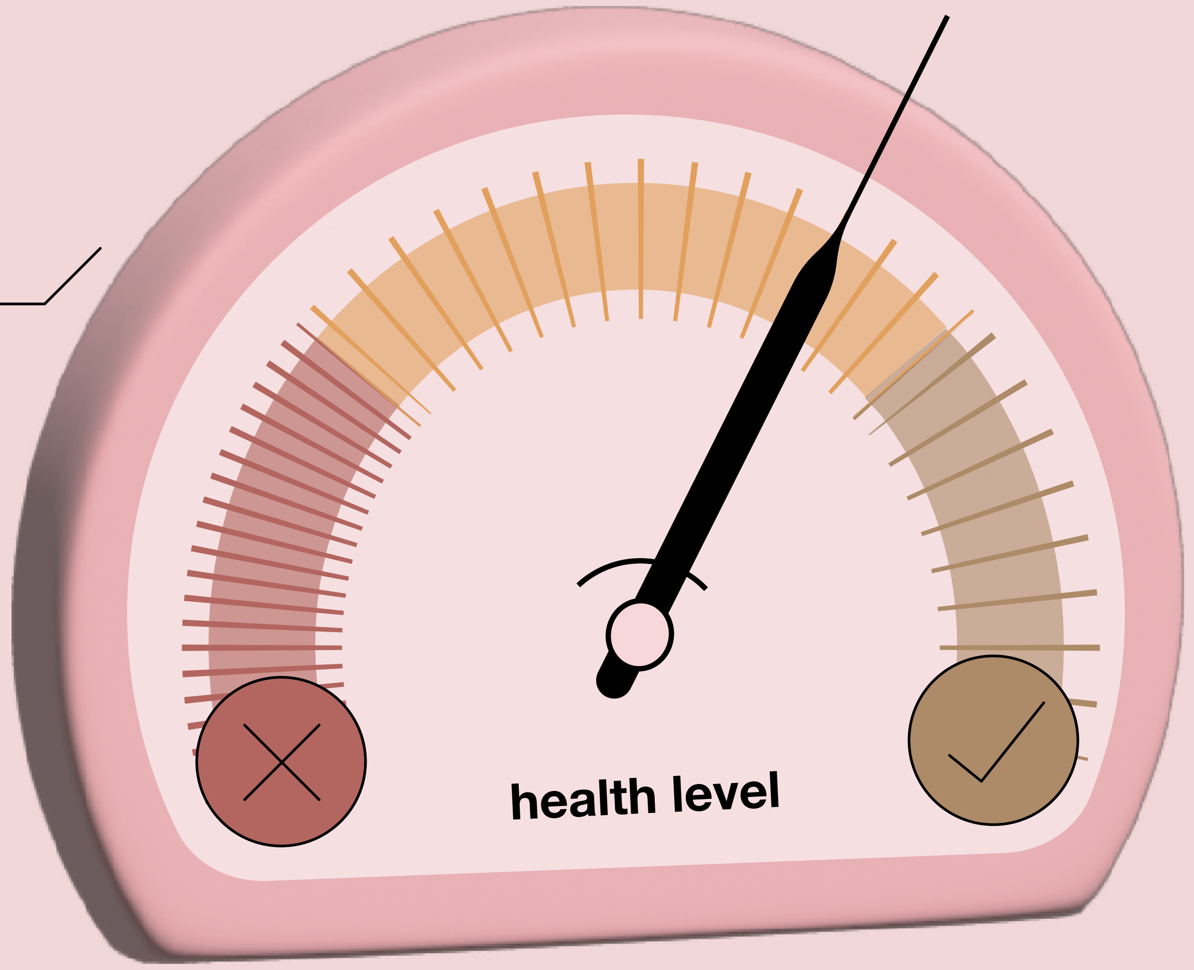
product experience

code quality



development processes

your score ? pt /38pt



0–15 pt
serious issues

Looks like your tech foundation has some serious issues which are severly hurting the core of your business, leaving their marks on your performance and reputation.

It's unlikely that there will be an easy fix. You'll most probably have to rethink your whole development process and refactor some parts of your digital product. A safe next step would be hiring an external tech advisor and bring in a fresh eye to thoroughly audit your tech foundation and help you identify clear next steps.

16–30 pt
danger zone

Although your software has some healthy building blocks, there are quite a number of underlying issues that will sabotage your business ambitions.

It is time to weed them out! Make a list of your weak points, and create an action plan with your technical team on how to fix them!

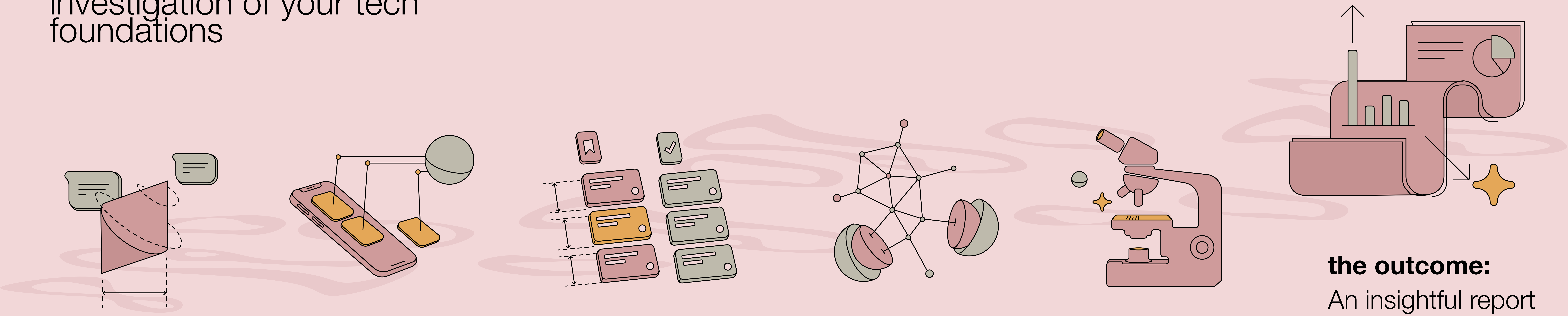
31–38 pt
solid tech foundation

Congrats, it seems like your tech foundation is in the healthy range.

Still, being cautious never hurts, right? You might need a deeper investigation on the questions where you didn't score the maximum 2 pt to identify what causes the issues.

are your results concerning? Supercharge is here to give you expert advice on how to improve your tech

our scientific process is made for a thorough investigation of your tech foundations



0 understanding your business context & goals

1 evaluation of your app’s production version

2 mapping out your development processes

3 checking your code quality with automated tools

4 conducting an expert code review

the outcome:

An insightful report that contains an objective expert assessment with improvement areas clearly defined.

we are Supercharge, the product innovation partner of businesses who want take control of their digital future.

We are obsessed with efficiency. Our 100+ engineering team built hundreds of digital products in the past decade, perfecting our development methodology to deliver robust software.

“We believe that better product quality results in happier users – and stronger businesses.

We created this evaluation sheet to let you build on our experience. We believe that objectively assessing the software quality of your own digital product will effectively help you make better decisions in the future.”

David Kovacs
CTO of Supercharge

do you
want to
dig deeper
and find
solutions
to your
underlying
tech
issues?

we’re here to help.
hello@supercharge.io

SUPER
CHARGE