

# CUSTOMLYTICS

**eBooks**

## Experiment Management Framework

A step by step guide for app store experiments by Customlytics

# Contents

Introduction	01
Phase 1 – Preparation for an experiment	02
Step 1 – Research	02
Step 2 – Hypothesize	03
Step 3 – Elaborate	04
Step 4 – Brief	05
Phase 2 – Executing the experiment	06
Step 5 – Produce	06
Step 6 – Test	06
Step 7 – Measure	07
Step 8 – Handle the results	07
Step 8.1 – Scale	08
Step 8.2 – Scrap	09
Step 8.3 – Adjust	10
Conclusion / Example	10



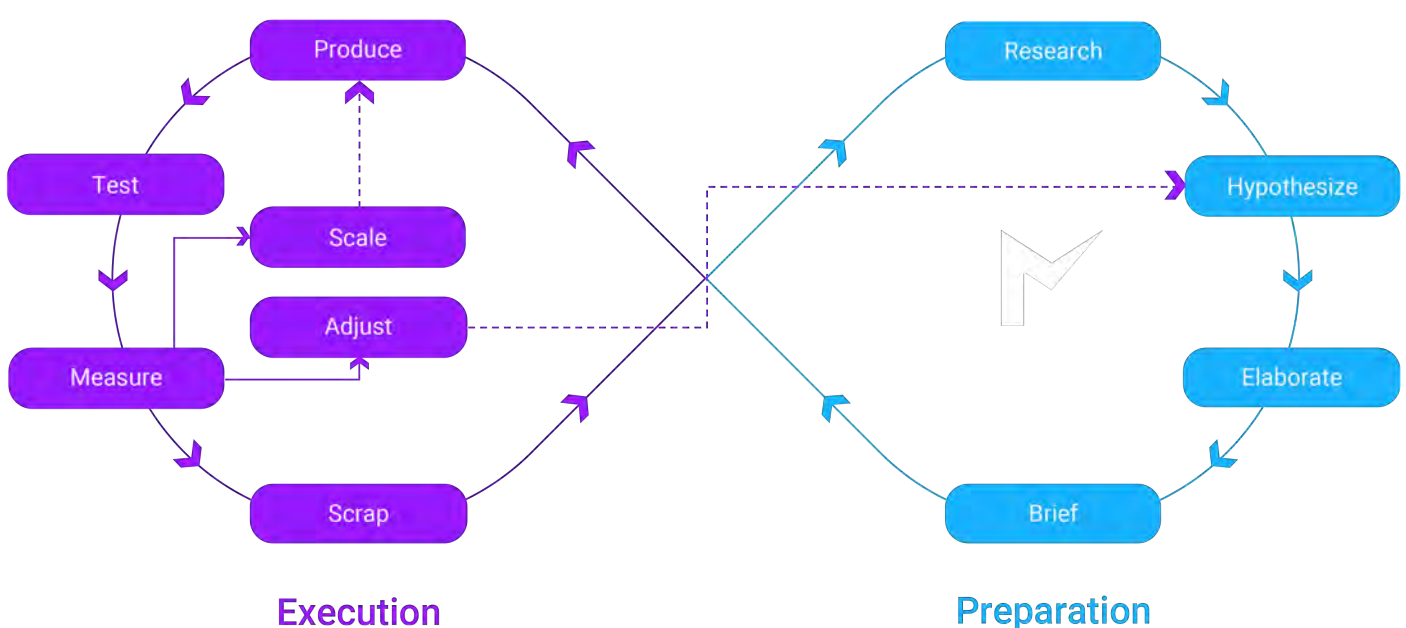
## Introduction

App store experimentation or A/B testing is one of the most vital elements of ASO due to its role in conversion rate optimization (CRO), which helps in increasing the conversion rate (CVR) from Store Listing Visitors or App Impressions to Installs. Yet, we all too often rush into testing for quick wins, i.e. XX% in CVR uplift, then forget about them so we can move on to the next test, leaving behind the chance to properly learn from the result of each experiment. We usually keep a “tweaking” mindset which is unsustainable. How long can one tweak their app store elements for growth? How much can one learn from the results of their experiments if their main aim is short-term tweaks? People just do it, they don't plan the tests, they don't organize the resources and team, they don't prioritize the test ideas, and so on. The result: They easily run into the errors of testing the wrong thing, in the wrong way, at the wrong time, to the wrong people, with the wrong platform, and their growth loop falls apart.

We need to think long-term with experimentation and use it as a healthy learning platform for sustainable growth, because experiments work best for idea validation, not conversions. We need a system that:

- forces us to set up a flawless testing environment before experimenting in order to reduce the risks of errors;
- reminds us to aim for learning and wisdom instead of quick results; and
- supplements our tactical, tweaking mindset with a strategic one.

For this reason, Customlytics proudly introduces a *more sustainable* way to do A/B testing: With a model that balances incremental improvements with strategic management.



Our model, namely the **Customlytics Experiment Management** framework, provides a step-by-step guiding system throughout the entire experimentation process. Each of the steps in the framework is designed to help you setup an error-free testing environment, as well as creating a virtually infinite loop of CRO, by providing:

- High degree of consistency across traffic channels, app store assets, etc.
- Long-term vision over a complete picture of organic growth
- Clear and strong sense of priority
- Healthy and wealthy pool of test ideas
- Wisdom and proper learning instead of hollow facts and figures
- A systematic and scientific method to manage the experiments end-to-end

Compared to what is publicly available in the market as of this moment, these traits have already made our model the industry's *most advanced*.

### This is how it works:

The model runs in two phases: Preparation and Execution

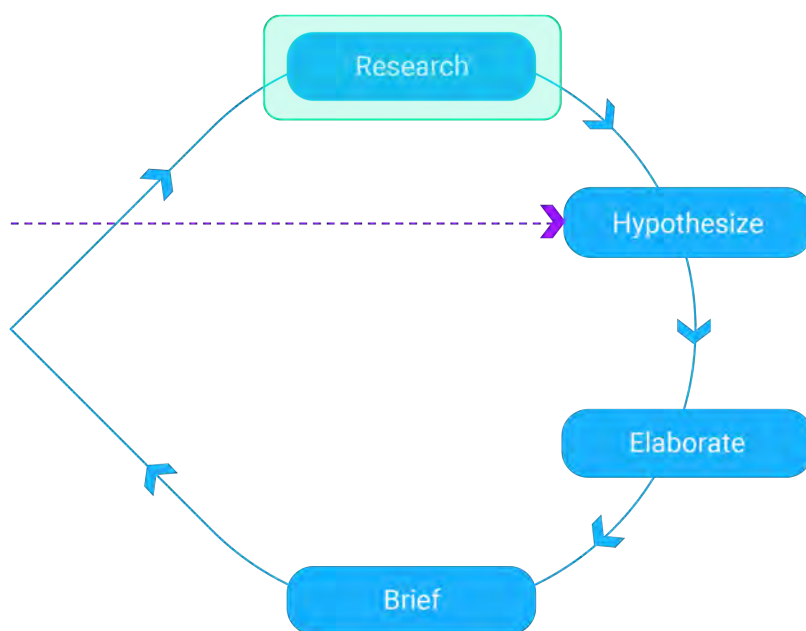
Preparation generally precedes execution, and execution follows standardized protocols so that the entire team can be on the same page and everyone knows what to do. It also leads back to the Preparation phase to fuel the next tests in a long-term roadmap.

### Phase 1: Preparation for an experiment

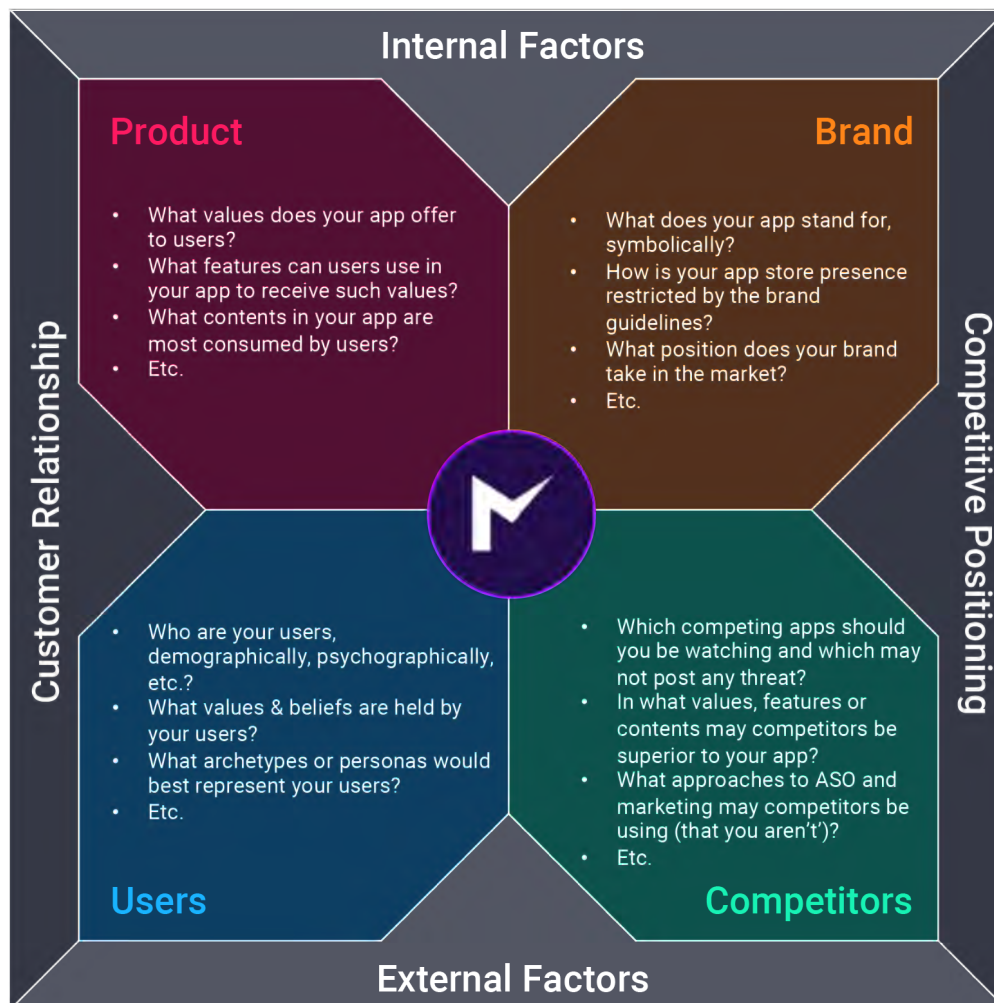
There are four things to do to get your experiment ready for testing:

#### Step 1 - Research:

The very first thing to do before anything can begin is to do thorough background research.



For a mobile app or game, this means you need to conduct research into 4 primary elements: the brand, the product, the users and the competitors.



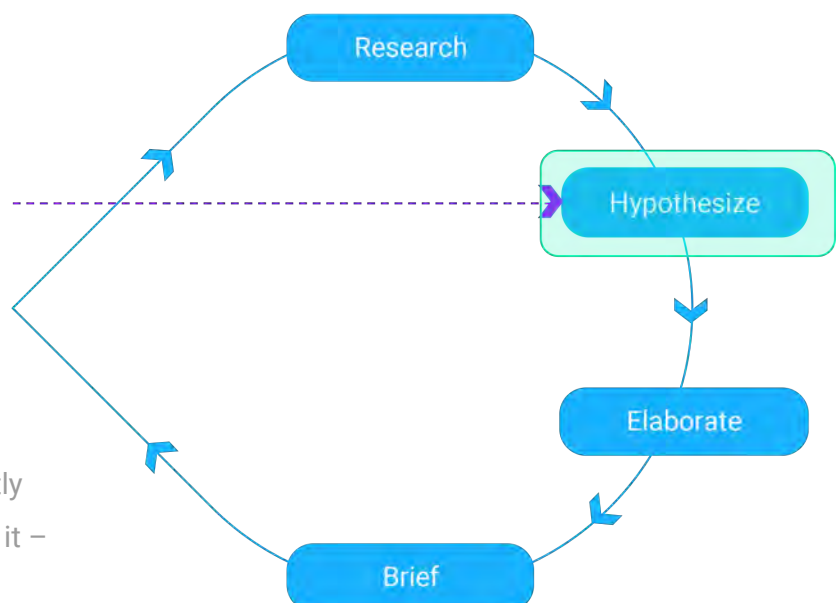
Example of an ASO research matrix developed by Customlytics.

The goal in this step is **data**; get as much data as possible and be well-informed.

## Step 2 - Hypothesize

Once the research is done and you have results that could inform you enough to generate test ideas, move on to formulate a hypothesis.

A hypothesis is required because it makes you generate an expectation, and subsequently use the test results to either confirm or reject it – hence validation or invalidation.



Not all hypotheses are strong enough. A strong hypothesis needs to be:

- Falsifiable: The only way to make sure you will have evidence to validate or invalidate a hypothesis is to make sure it is falsifiable and not based on faith.
- Structured: A hypothesis is strong when it states the variable or action, its expected result, and the reason or rationale behind it.

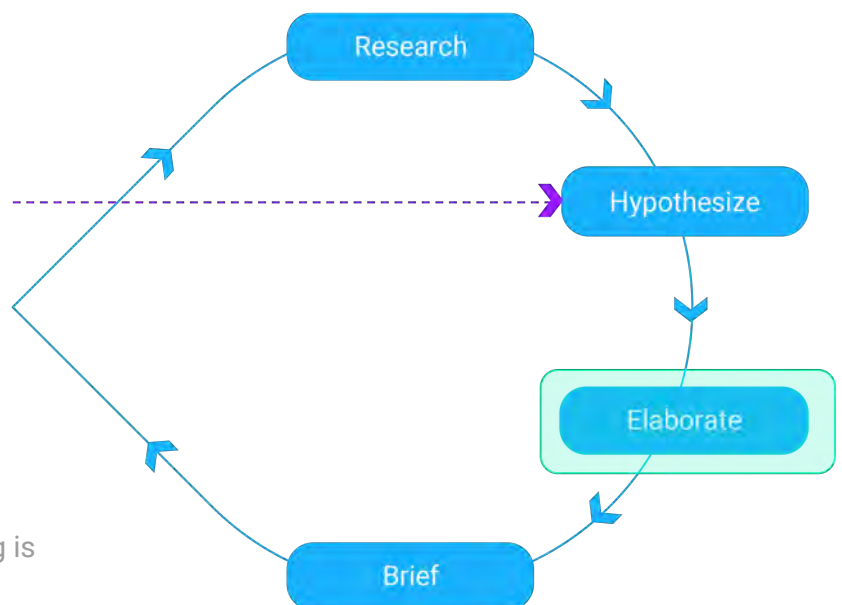
**"IF \_\_\_\_\_ , THEN \_\_\_\_\_ DUE TO \_\_\_\_\_ ."**  
[Variable] [Result] [Rationale]

When you can expect this – *"if something happens then something else happens as a result due to a specific reason"* – then you're on a good way to learning what you're testing, what its result is and why.

### Step 3 - Elaborate

Even a strong hypothesis on its own is not enough to set up a great test. Elaborate it into something of higher value instead.

Even a perfect hypothesis doesn't necessarily communicate what you're testing, why, how, when to test, who runs the test, how long it should last, and so on. You need **strategic planning** where everything is defined in advance.



You also need to make sure the artist who materializes your test ideas and hypothesis (in your head all along) shares your vision. For this, you need **creative production planning**, which has 3 components.

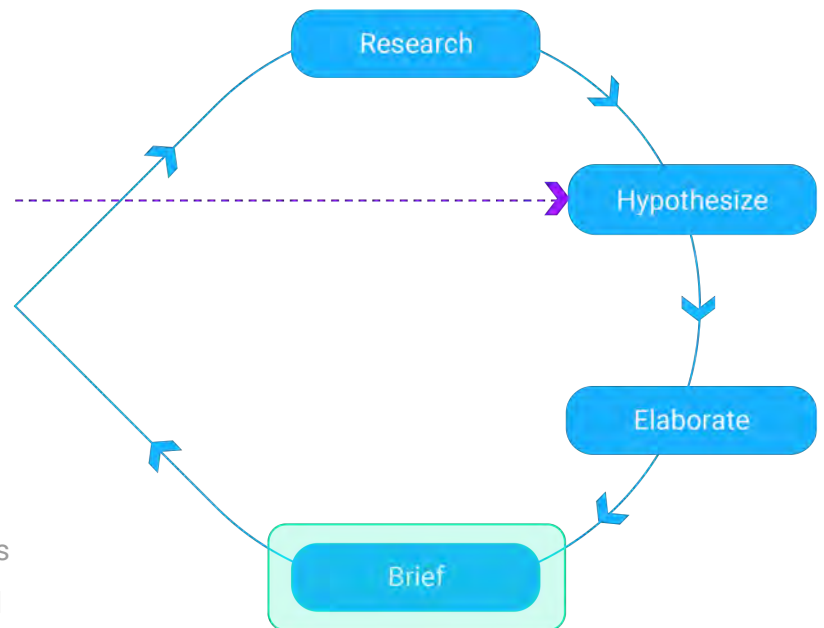
- **The story:** or the overarching concept behind your test (variants) and the core of the message you want to communicate
- **The storytelling:** or the techniques you want to use to convey the message, either visually or textually, or both
- **The storyboard:** both the story and storytelling techniques are still very abstract, so you need a storyboard that summarizes their tangible product.

With elaboration, your hypothesis has the chance to become a whole **strategy** to test. It could be generic or unpolished, but you can tweak it later down the road. The goal, for now, is to have a general **creative direction** that, once validated, can lead to a whole host of experiments to come for months, if not years.

#### Step 4 - Brief

Once the storyboard is developed, it's time to make a creative brief to instruct your artists over the materialization of the test variants.

Feel free to be playful and creative here if it helps, but don't forget that ASO also requires reasons or logic. Detailed specifications such as the size and resolutions of the screenshots, the quantity, or the OS and device model you're optimizing for, for instance, should be stated clearly so the next person knows what to do.



For example:

- iPhone 8: 10 screenshots, 1242 x 2208 pixel resolution
- iPhone XS MAX: 10 Screenshots, 1242 x 2688 pixel resolution
- iPad: 10 screenshots, 2048 x 2732 pixel resolution

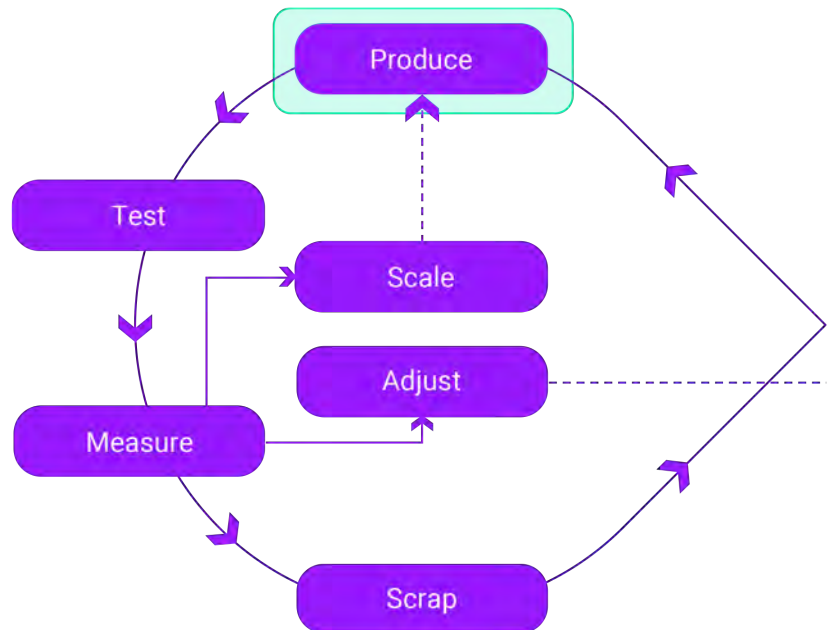
The details may vary – copywriting, graphic design and animation projects are different, after all – but make sure they're all comprehensively specified.

## Phase 2: Executing the experiment

If you've reached this phase, it means you have everything your creative team needs to produce the test materials: the storyboard and the creative brief. Now you're ready for the execution.

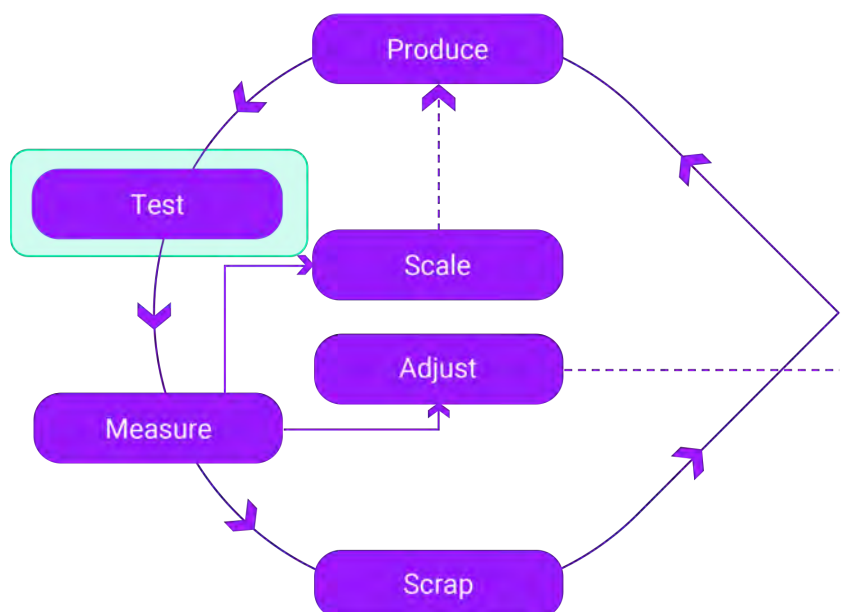
### Step 5 – Produce

It's simple. You have what to create and how to create it, so production can begin accordingly.



### Step 6 – Test

After your test ideas have been materialized into a specific kind of app store asset, you can think about running the test now.



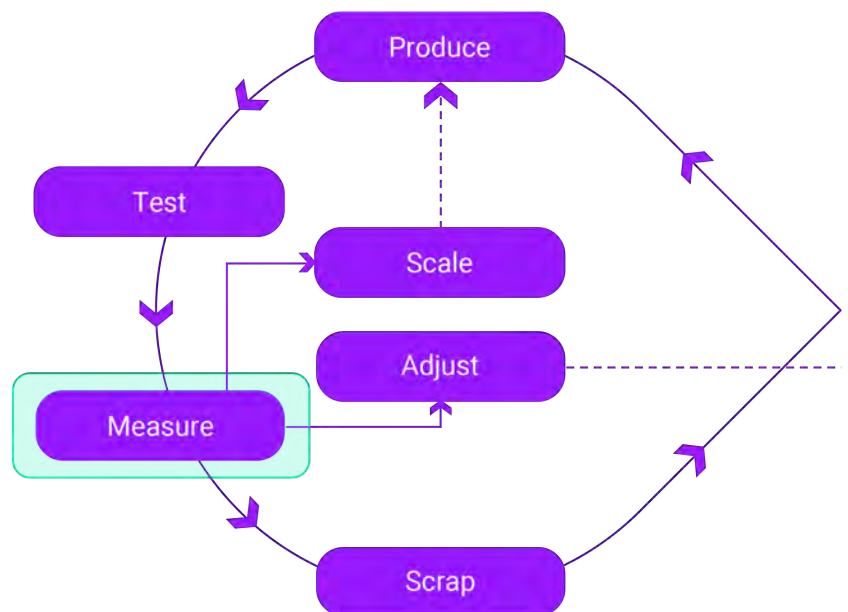


Here's how to do it properly:

- **Pick the right testing method and platform:** there are true experiment methods on platforms like the Google Play Store Listing Experiments, and then there are quasi-experiments like Apple Search Ads Creative Set Testing, which aren't experiments per se, but they help you validate certain forms of hypotheses.
- **Deploy** the materials of the test variants to the platform of choice in accordance to the requirements of the method of choice.
- **Execute** the test by allowing the platform to serve the test variants to the users.

## Step 7 – Measure

Running the test for a while (one week or one month, for example, depending on the confidence intervals), you will receive a form of result – either a win, a loss or a tie – so measure it, and be smart about it.



You can do this by:

- Identifying the **differences in conversion rates** (CVR) between the Control and the test variants
- Either **validating** or **invalidating** the hypothesis behind the test, using the result
- Making sense of the result for **proper learning** – find an interpretation of it, or a guidance for the next tests

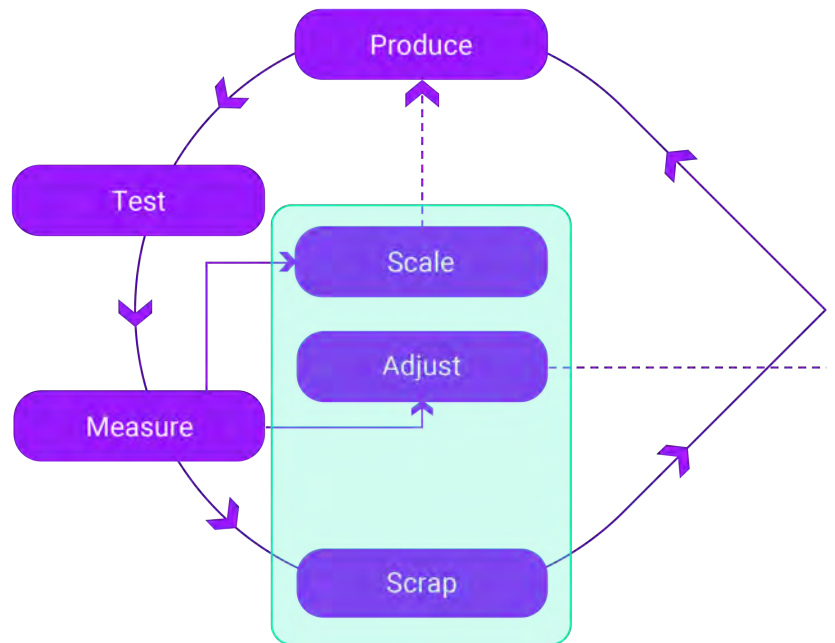
The goal is to make sure you learn something – anything – but in the right way.

## Step 8 – Handle the results

Our model comes with a set of **protocols** for the treatment of a test result that are designed to keep your team on the same page and everybody knows what to do with a test variant when the result comes in.

There are 3 protocols to follow at this stage:

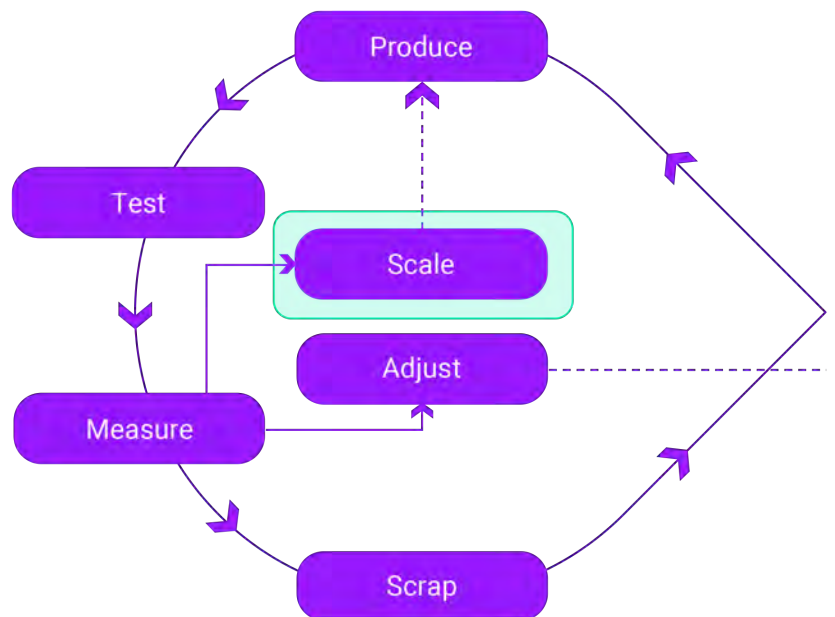
- Scale the strategy or variant to grow
- Scrap it to reset the process
- Adjust it to optimize



### Step 8.1 – Scale

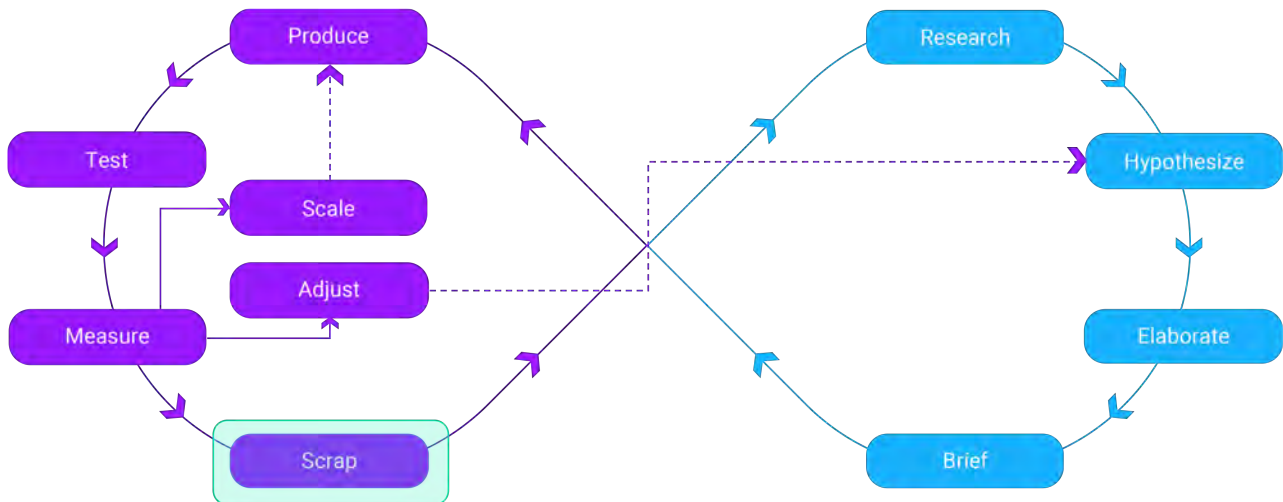
This is a typical action to take for the “champion” type of test variants, i.e. very convincing win over the control variant. You can scale it by expanding it to more languages (localize it), or stores (Play Store to App Store), or assets (from screenshots to the feature graphic). Make sure you maintain the same strategy and don’t take anything for granted – keep testing.

The scaling starts from production – producing materials for the same family of tests – followed by more testing, measuring, and so on. The fact that you’re able to scale here means you already have a **growth loop** at this stage.



## Step 8.2 - Scrap

This is a typical treatment for a very poorly performing test variant. It usually shows little to no value for growth. This means you should scrap it and take a step backwards to identify what else you can test.



You have two options:

- Revisit a previous step in the Preparation phase to check if you still have other hypotheses or storyboards lying around somewhere – use them
- Restart the whole process if there is nothing more you can test – do the research again for a clean slate

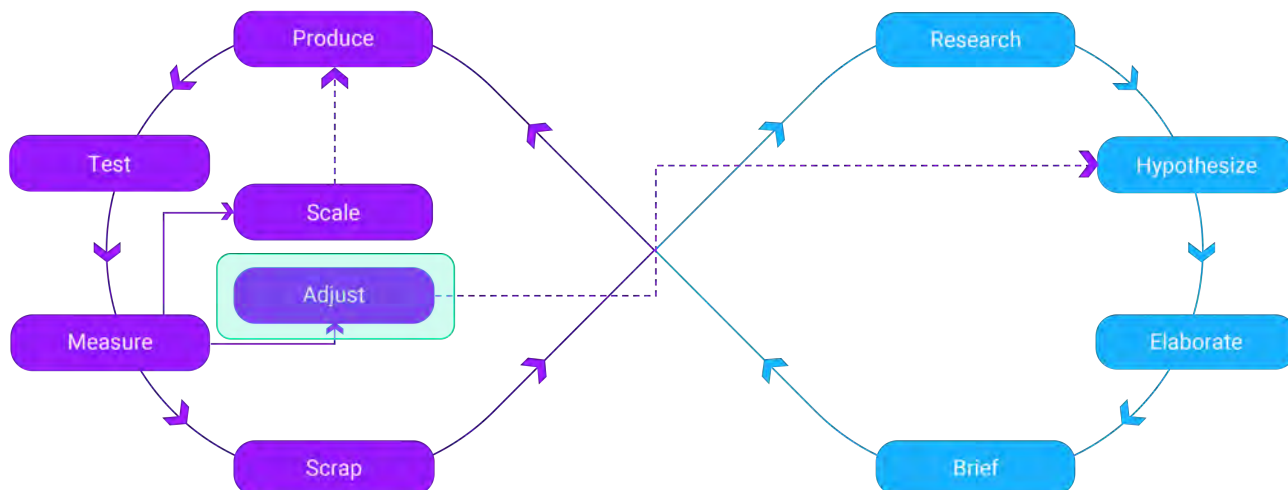
Again, the idea is validation, not conversions, so make sure you learn something helpful. If the loss helps you figure out a specific strategy that you should NOT use for CRO, it's a **win** for your whole ASO game.

Also, the better you have elaborated on your hypothesis earlier, the less likely it is for you to have to restart everything. Maybe you just need to add a few details in the creative brief, maybe only the wording doesn't work, etc.

In the end, the variant that you have to scrap could teach you something. For this reason, it represents your **feedback loop** in CRO.

### Step 8.3 - Adjust

Finally, there is everything in between – a win with space for improvements, a loss with possibility for redemption, a tie that requires further clarification – and you just need to change ONE detail to continue testing for incremental improvements.



This is where iterative testing can be done, to a point where only the details matter and *tweaking makes more sense than strategic testing*.

Remember to formulate or reformulate your hypotheses before testing. The framework will generally lead you back into the *Hypothesize* step in the *Preparation* phase, for this reason.

### Conclusion

Above was a very brief introduction into our model. In practice, it entails much more detail of both strategic planning work and the actual operational level activities – including reporting. They could easily be hidden away by an unsustainable approach to experimentation – make sure you won't fall into this trap.

Read more about our most recent comprehensive [case study of COUP](#), one of our clients, which is available on [our website](#). TLDR? The whole story is like this:

- We used the model to develop **4 creative directions** for the app to test initially
- One of them showed superiority with CVR uplift of **up to 23%** in the form of a screenshots experiment
- We then scaled it to **3 more** app store **locales**
- At the same time, we began testing adjusting it for iterative testing, with a roadmap until the **next year** based on over **30 testable elements** to choose from.



- Every move we made, we only needed to plan for it **once**, in the very beginning, because we followed the model – and **never had to second-guess** ourselves.

The collage consists of several promotional banners for the COUP app:

- Top Row (English):**
  - Discover the smarter way to travel (with scooter image)
  - Open the app to find a COUP nearby! (map view)
  - Enjoy the ride not the traffic (booking completed screen: 14 min, 2,94 €)
  - Join the future of urban mobility (woman on scooter)
  - 24/7 support putting you first (support chat screen)
  - Find your new ride today (woman on scooter)
  - Stay in control & track your trips (trip history screen)
  - One login for multiple cities (city selection screen)
- Second Row (English):**
  - Book within seconds (scooter icon)
  - Always near you (phone with map)
  - 1 app, 3 cities (Berlin, Paris, Madrid icons)
  - Free reservation (map with reservation point)
  - Step by step assistance (phone with instructions)
  - Helmet & hygiene cap included, always (woman wearing helmet)
  - Check your rides with one click (phone with booking ended screen)
- Third Row (German):**
  - 3x10 Min Kostenlos (Promo Code: DOWNLOAD)
  - Mach noch heute deinen ersten Ride (scooter)
  - Kostenlos reservieren und stornieren (woman on scooter)
  - Behalte deine Fahrten im Blick (phone with trip list)
  - Genieße die Fahrt, nicht den Stau (phone with booking screen)
  - Unser Support ist 24/7 für dich da (phone with support chat)
  - ...ideal, wenn der Platz knapp ist. (Ruhige Refuge statt Redaktionsbüro) - stern (man on scooter)
- Bottom Row (English):**
  - The hassle free way to travel (scooter)
  - Always there for you (map with many scooters)
  - Free reservation (phone with reservation screen)
  - Step by step guidance (phone with instructions)
  - Powerful batteries electrified journeys (battery icon)
  - Fastest way from A to B (man running with scooter)
  - One app, one ride, multiple benefits (phone with various app features)
  - Easy, safe & clean (man on scooter)

*We test strategies first and tweak second*

Our model gives us and our clients an error-free setup, super management workflows, vast amounts of test ideas, healthy KPIs, and peace of mind. App store experimentation doesn't get much easier than this. It can be this easy for you as well!

Drop us a line at [info@customlytics.com](mailto:info@customlytics.com) and let us help your app excel in CRO.

