

Experimentation Program Toolkit

To help kick-start or accelerate your experimentation program

by Rebecca Bruggman

Welcome to the Experimentation Program Toolkit	3
Section 1: Setting Up your Program Structure & KPIs	4
Experiment RASCI	6
Program-Level Documentation	7
Goal Tree: Template & Example	9
Section 2: Build, Launch & QA Experiments at Scale	10
Build & Launch a Web Experiment	12
Build & Launch a Full Stack Experiment	16
Creating New Entities for Optimizely Web	19
Section 3: How to get Experiment Ideas from Everywhere in your Organization & Creating your Roadmap	24
Experiment Roadmap	26
Ideation Session Problem & Solution Mapping	27
Experiment Idea Scoring Framework	28
Section 4: Reviewing and Prioritizing Experiments	29
Experiment Review Meeting: Agenda, Notes and Backlog	31
Experiment Brief	33
Section 5: How to make Experimentation Visible Consistently Across your Organization	35
Experiment Start Email Template	37
Results Share Email Template	39
Visualizing Experimentation Program Examples	41
Section 6: Demonstrating ROI	42
Experimentation Program Quarterly Update Presentation	44
Quarterly Experiment Program Update Email	45

Welcome to the Experimentation Program Toolkit

Hello Experimenter! As Optimizely's Experimentation Program Manager since 2018, my job is to make sure we are "drinking our own champagne" and truly running a best-in-class experimentation program, lovingly referred to as "Optimizely on Optimizely." To do this consistently requires multiple tactics over the course of days, weeks, months and years.

My team and I have created blogs and documentation on the importance of having an [Executive Sponsor](#), how to keep experiment ideas flowing by hosting [Ideation sessions](#) and how to think about driving your [KPIs with metrics movable via experimentation](#). However, one of the key areas of investment I have made that's been extremely impactful is documentation for all our processes.

A program manager (me), organizational buy-in, proper enablement and alignment on key metrics are all critical to the success of a mature experimentation program, but having all the processes documented for running a program day-to-day and quarter-over-quarter are also essential for ongoing success.

Comprehensive documentation keeps stakeholders aligned and frees me, and others across the organization, to focus directly on building the most impactful experiments and getting their hypotheses, metrics and set up correctly.

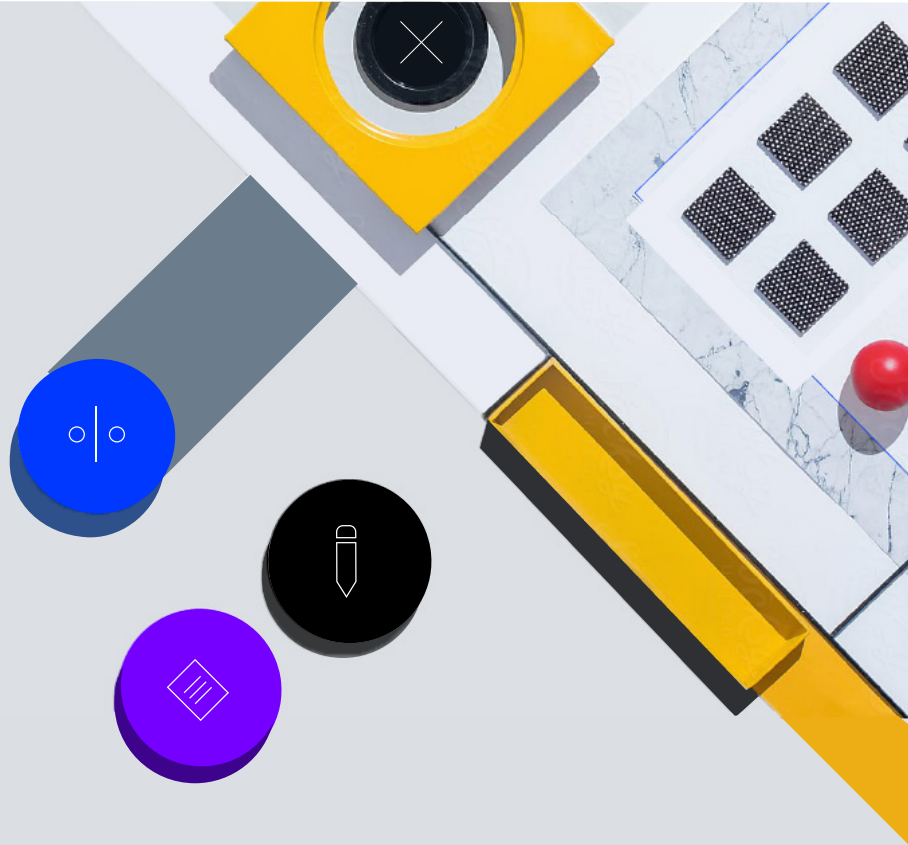
Over the years, I've developed hard-won insights about how to run the right kind of program -- and created a set of templates and tactics that help maintain a program through highs and lows. I'd like to share all these insights by sharing this toolkit with you.

I've taken all the templates I use internally for Optimizely on Optimizely and made them into this toolkit. This can be used by anyone who's looking to set up, expand or increase the ROI of their experimentation program.

I hope these support you in building and evangelizing your experimentation efforts!

Happy Experimenting,
Rebecca Bruggman
Program Manager, Optimizely on Optimizely

PS - if you have any feedback or follow-up questions, please reach out to me directly on twitter @bexcitement.



Section 1

Setting Up your Program Structure & KPIs

Let's start by asking the right questions. When starting an experimentation program, find out:

- Who are my key stakeholders? Who will contribute to experimentation?
- What structure will my team have? Will it be centralized or dispersed?
- What are the key metrics I'm looking to impact? Where are the areas of my business where I want to accelerate learning?
- How will I onboard new people to experimentation?

An experimentation RASCI (stakeholders who are defined as Responsible, Accountable, Supportive, Consulted, or Informed) is a great starting place to see who needs to be involved in an experiment launch and the program overall. Below, I've included a template to help you map out your RASCI.

From there, start thinking about who you have on your team and what structure would make the most sense for you and your team. There are a number of different program structures you can use depending on how centralized your experimentation efforts need to be and how your team is resourced, which is outlined fully by Optimizely's lead Strategy Consultant, Alek Toumert [\[here\]](#).

At a high-level, some of the considerations when selecting your program structure include:

- How are experiments prioritized and built?
- What level of control is needed over experiment velocity?
- What dedicated resources do you have for the program?
- How much bandwidth is available for ongoing enablement?
- How many teams need/want to be involved?
- Is there buy-in from key stakeholders to prioritize experimentation against other feature work?

Once you've thought about the stakeholders and contributors to your program, what structure works best and added all that to the program documentation included below, you can map out a goal tree! A goal tree is important to align with your team on to identify the metrics you can move through experimentation that track back to your top line key performance indicators (KPIs) [such as revenue or sales accepted leads (SALs)] and can go a long way to show the impact of your program. A goal tree can ensure you are focusing on the right metrics, especially when you are first starting the program. Starting with experiments that have the potential to garner a lot of learnings toward the most important metrics for your organization is a great beginning for experiment ideation.

I've included a goal tree template and an example completed goal tree below for you to get a visual representation of what these look like and how to build them for your own team.

We also have a [great post on our Knowledge Base](#) covering even more details on how to build a goal tree.

Experimentation Program RASCI

 Download Template



Experiment RASCI

Experiment Stage	Responsible	Accountable	Supporting	Consulted	Informed
Ideate -Providing ideas to backlog -Utilizing relevant data points					
Prioritize -Scoring ideas along set criteria -Final determination on what gets executed					
Test Design -Audiences, metrics, sample size, etc. -Test plan creation					
Creative Design -Develop relevant assets for variations					
Variation Build -Develop relevant code to test plan					
Optimizley Implementation -Set audiences, metrics, etc. within platform -Set relevant feature variations and keys					
QA & Launch -Whitelist user IDs for experience testing -Approve and hit 'Start'					
Results Analysis -Determine success and learning(s) -Distribution of results to stakeholders					

Experimentation Team

Example Team



Claire Vo
CPO, Product



Becca Bruggman
Staff Program
Manager



Michael O'Connell
Director,
Demand Gen



Takeshi Young
SEO Lead



Bryce Wellington
Staff Marketing
Engineer

Key Stakeholders:

Executive Sponsor: {name}

- What does this person own? How can they help?
- How to contact them?

Program Manager: {name}

- What does this person own? How can they help?
- How to contact them?

Engineering Champion: {name}

- What does this person own? How can they help?
- How to contact them?

Other Key Contributor: {name}

- What does this person own? How can they help?
- How to contact them?

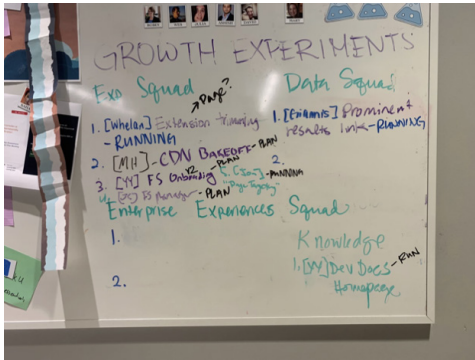
Slack Channels:

- Name of Slack Channel {linked}

Current Experiments:

- Link to your backlog {linked}
- Link to any monitoring dashboards for experimentation
- Picture of visual spaces that experimentation is represented

Example Visual Representation of Board:



Meetings:

{Meeting where Experimentation is discussed}

- When and where does this meeting happen?
- Who owns this meeting?
- What happens in this meeting?
- Why should people come?
- How to get on the agenda for this meeting?

{Other meeting where Experimentation is discussed}

- When and where does this meeting happen?
- Who owns this meeting?
- What happens in this meeting?
- Why should people come?
- How to get on the agenda for this meeting?

Documentation for Running an Experiment:

- [Link to documentation {linked}](#)
- [Link to documentation {linked}](#)

Blog Posts:

- [Link to Blog Post {linked}](#)
- [Link to Blog Post {linked}](#)

Presentations:

- [Link to Presentation {linked}](#)
- [Link to Presentation {linked}](#)

Getting Started Materials:

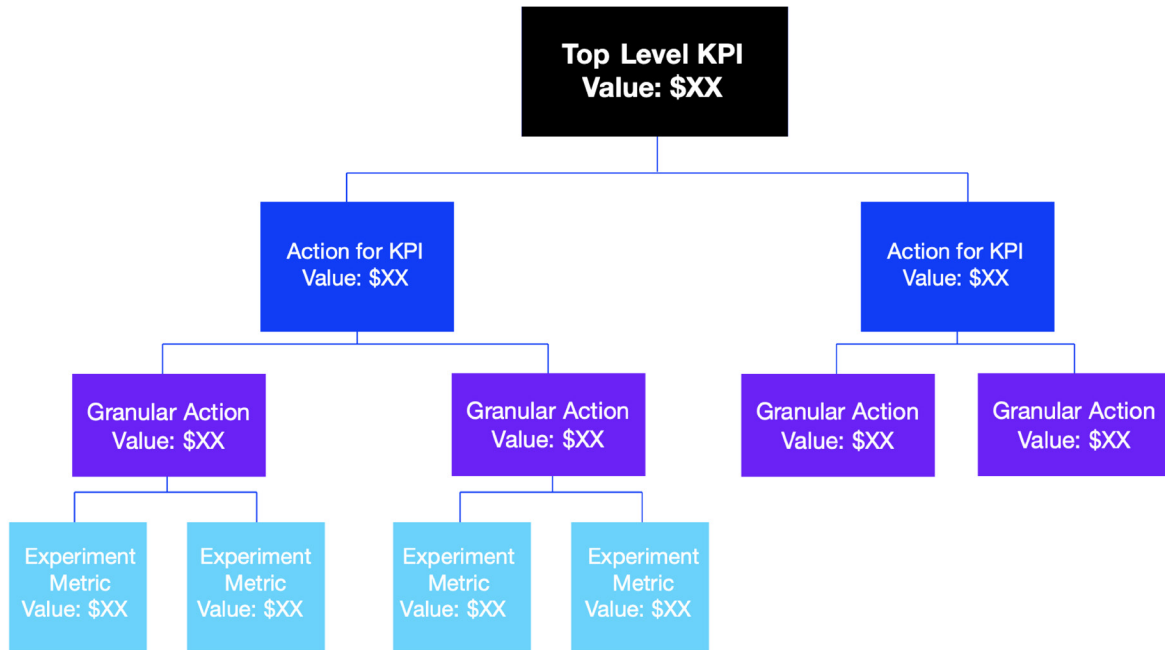
[Optimizely Knowledge Base](#)

[Optimizely Academy](#)

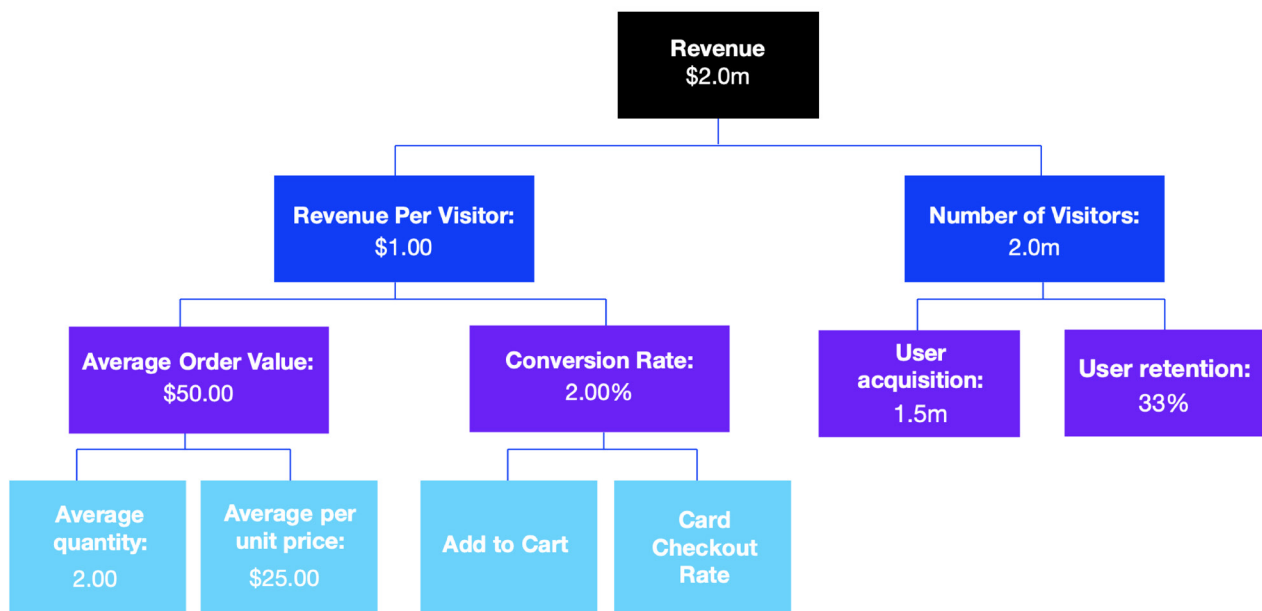
Goal Tree: Template & Example

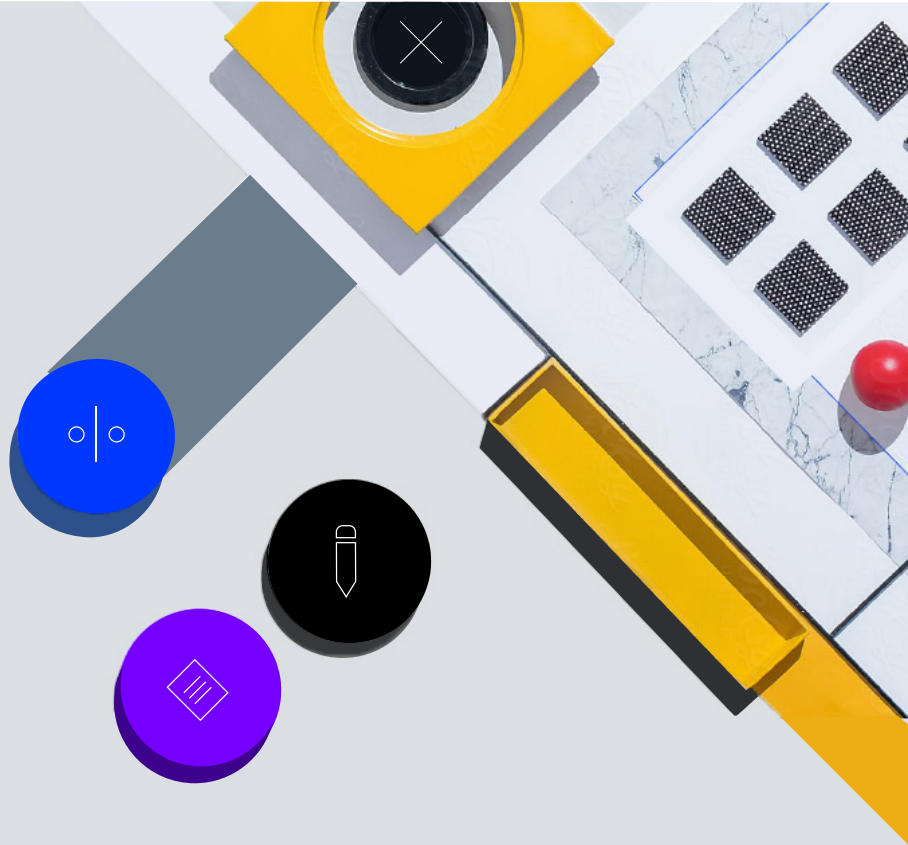
 Download
Template

Goal Tree Template



Example Goal Tree





Section 2

Build, Launch & QA Experiments at Scale

When we were working to formalize our processes and program at Optimizely, we quickly realized how important it was to have comprehensive documentation that all key stakeholders had reviewed and aligned on. Not everyone loves process, but good process can help free up teams in a few key ways:

- Having an impact through their experiments and being creative as opposed to having to figure out each time what steps are needed to safely and successfully launch an experiment.
- For onboarding, having the experimentation process documented is basically the equivalent of scaling me and reduces the number of 1:1 sessions with the team. (And as a human, I don't scale.)
- Finally, if anyone wants to make modifications to the process, there is a single source of truth for where those changes can be discussed and made.

Below, you'll find templates to document:

- Running a client-side experiment (we use Optimizely Web) end-to-end
- Running an experiment in code (we use Optimizely Full Stack) end-to-end
- Creating new entities (pages, metrics, audiences, etc) in an Optimizely Web project

Keep in mind that it takes time for a team to get accustomed to leveraging new processes consistently. You'll also need to update your processes based on team feedback and the shifting needs for your program. With a single source of truth, only one place needs to be updated and re-socialized if it becomes clear a process needs to be updated or removed.

For each of these, it is helpful to get sign-off from your key program stakeholders, end users of the process and anyone who would be impacted by the process such as code reviewers. These are living documents. I recommend reviewing once per quarter to ensure they are still relevant from a process and stakeholder perspective.

Build & Launch a Web Experiment



{Shortlink for this documentation}

STOP

Looking to build or update an Optimizely X Web Entity (ie. Page, Audience, Metric)?

This document talks about building an Optimizely X Web Experiment or Campaign and getting it reviewed. For reusable components (e.g. Pages, Audiences, Extensions, etc.) creation or updates, check out [{this link}](#).

Looking to run a Full Stack Experiment?

[{this link}](#)

What is a Web Experiment?

A web experiment is an experiment on our website. These include Web A/B Tests and Personalization Campaigns on [{this website}](#) or [{this website}](#).

This document focuses on the steps needed to run a Web experiment. For info on running a Full Stack Experiment or Rollout, go [{this link}](#).

Web Experimentation Steps

Every web experiment should go through the following steps:

1. Plan (Idea Stages: “Requirements”, “Creative”)

- a. Decide whether you should run an experiment or rollout for your change.
- b. Create an Idea in the backlog [{link to your Optimizely Program Management Team or Experiment backlog}](#)

(if you don't have access to the above, please ping [{program manager}](#))

- Include a hypothesis, primary metric, monitoring metrics and variation screenshots.
- Add [{Experimentation Team}](#) and other relevant team mates as “followers”. **(or tag them in your idea backlog in a spreadsheet)**

1. As a follower, you should Score the idea and add a Comment if there is anything you think should be added to the experiment, such as a monitoring metric or different audience.

- c. Add tags or filters to the idea:
 - Tag for teams:
 1. Ex: #demand-generation, #product
 - Tag for okr:
 1. Tag for Quarter: ex: Q4FY19
- d. Post the Idea link in **#team-slack-channel** and schedule a time to present at **Experiment Review Meeting** by pinging {Program Manager}.

2. Implement (Idea Stages: “Development”, “Setup/QA”, State: “Active”)

- a. Create an experiment and “link” it to your Program Management Idea (**or add to experiment roadmap spreadsheet**)
 - If you need support building an experiment, please **{follow these steps to get technical support for experiment build}**.
- b. Add the URL for your idea in Program Management (**or experiment roadmap**) in the description of the experiment.
- c. Web experiments must undergo an External JS Review (see below for details on requesting and performing a technical review) before going live
- d. Any experiment which changes the UI requires a design review before it goes live. If you need help with design to prepare for experiment review, submit a request **{link to request process}**.

3. Run (Idea Stage: “Testing”)

- a. Please communicate about your experiment launch at least 24 hours in advance:
 - Add a comment to your Idea giving followers a heads up when the experiment is scheduled to start
 - If your experiment will be visible to customers, send an email to **team@yourcompany.com** detailing your hypothesis and variations
 1. Visible Changes template email: **{link to template}**

4. Analyze (Idea Stage: “Analysis”)

- a. Schedule a slot in **Experiment Review Meeting** to go through your experiment results
- b. Record your results in Program Management (Record Results & Add Notes) (**or experiment backlog**) and share with the team via **team@yourcompany.com**
 - Visible Changes Results Sharing template email: **{link to template}**

5. Rollout

- a. Depending on the experiment and results, this may include moving [100% of traffic to the winning variation](#), removing experiment code from your codebase, or implementing a “web” variation in code. You should discuss with your team ahead of time and be prepared to walk through your rollout plan during Experiment Review!
 - If you are leaving your experiment as a hotfix for a certain amount of time, log a JIRA ticket with the engineering team that would need to incorporate it and label the experiment in Optimizely with [Hotfix]
- b. When you are ready to rollout, follow the **instructions for launching a rollout [here]**

6. Archive (Idea Stage: “Complete”, Status: “Archive”)

- a. Before you archive the experiment, make sure to implement in code the behavior you need.

7. When archived, the snippet as well as datafiles assume that no such experiment exists and so any reference to the experiment will result in the default behavior. So make sure to update the default behavior before archiving.

Technical Review and Approval for Web Experiments and Campaigns

Every Web experiment should undergo the following **External JS Review** Process before going live:

1. Set up and QA your experiment/campaign.
2. To request a review:
 - a. Create a JIRA task in the {**team that handles reviews**} project, including:
 - i. Title: [Code Review] {Experiment Name}
 - ii. Experiment ID/Name
 - iii. Link to the **Program Management Idea or Experiment Backlog**
 - iv. Description of any Custom Code Included in experiment/campaign
 1. Experiment Shared Code?
 2. Variation Custom Code?
 3. Page or URL Targeting Conditional Activation Custom Code?
 4. Project JavaScript?
 - v. Intended Audience
 - b. Add this JIRA URL to the “Setup & QA” notes in your Idea.
 - c. Label: Experiment-Review
 - d. Add {**engineering managers**} as a follower on the Idea in Program Management and request a reviewer and include the JIRA link in your comment. They will help identify an engineer for review.
 - i. As this is a code review, it falls under the **24 response SLA**.
 - ii. If a JIRA ticket for an experiment code review is un-responded to after a week, it will be treated as approved and launched.

How to Perform a Technical Review on a Web Experiment

Have you been asked to review an experiment before going live? Follow these steps:

1. Read through the JIRA and learn about the experiment.
 - a. If it looks complex (for example, if it uses cross-project metrics or special targeting), put some time on the calendar of the person who implemented the experiment and have them walk you through it end to end.
2. Review any custom code:
 - a. Experiment Shared Code
 - b. Variation Custom Code
 - c. Page or URL Targeting Conditional Activation Custom Code
 - d. Project JavaScript

3. Review Experiment Logic. Will this Experiment work as it is intended to? Here are some things to check
 - a. Page / URL Targeting Conditions
 - b. Audiences (does it match with intended Audiences in JIRA? Is there a test audience that you can force yourself into?)
 - c. Cross Project Metrics (are they implemented properly?)
 4. Review potential conflicts
 - a. Use **{Program Management or Experiment Roadmap}** to examine other Ideas with the same touchpoint(s) that are currently Active. Will any of them conflict with the proposed change?
 - b. Will the experiment conflict with an upcoming feature rollout inside our app?
 5. At this point, the Experiment Reviewer requests changes in the JIRA, comments, or accepts and marks the JIRA as “Done” when all custom code passes a code review.
 6. Make sure that the experimenter alerts the teams whose area of the app the experiment touches. For example, if the experiment is running on the **{place in app}**, ping the **#team-channel** when the visible changes email message goes out.
-

Additional Help

Where can I learn more about how to implement my experiment?

See the How to Build an Experiment in Optimizely X above

What are our naming conventions?

Web

- Experiment Name: [Owner First Name] Touchpoint Name - Idea Name
Examples:
 - [Whelan] Page Builder - Dynamic Websites Awareness (v2)
 - [Zach] Navigation - Program Management Plug
- Description: Link to Program Management Idea or other relevant documentation
 - If a hotfix, include JIRA link for hotfix implementation into codebase
- Variation Name: Description of Variation (please do not use default variation names like “Variation #1”--but do keep it short since these are included in the Web client payload)

Should I use this process for Personalization Campaigns too?

Yes.

What is the weekly Experiment Review meeting for?

We'll use Experiment Reviews to go through experiment plans and results together and provide feedback and advice to the teams running them. The goals of this meeting are to 1) ensure we run high-integrity experiments and campaigns and 2) spread knowledge and best practices around our team.

I have questions. Where do I go?

Please add questions directly to this document, or post them in

#experiment-team-channel or ping **{program manager}**.

Build & Launch a Full Stack Experiment



{Shortlink for this documentation}

STOP

Looking to run a Web Experiment?

{this link}

What is a Full Stack Experiment?

A full stack experiment or rollout is an experiment or feature management configuration that exists in the back-end code base. These include Full Stack A/B Tests, Rollouts and Feature Flags for {this website} or {this website}.

This document focuses on the steps needed to run a Full Stack experiment or Rollout. For info on running a Web Experiment, go {this link}.

Full Stack Experimentation Steps

Every full stack experiment should go through the following steps:

1. Plan (Idea Stages: “Requirements”, “Creative”)

- a. Decide whether you should run an experiment or rollout for your change.
- b. Create an Idea in the backlog {link to your Optimizely Program Management Team or Experiment backlog} (if you don't have access to the above, please ping {program manager})
 - Include a hypothesis, primary metric, monitoring metrics and variation screenshots.
 - Add {Experimentation Team} and other relevant team mates as “followers”. (or tag them in your idea backlog in a spreadsheet)
 1. As a follower, you should Score the idea and add a Comment if there is anything you think should be added to the experiment, such as a monitoring metric or different audience.
- c. Add tags or filters to the idea:
 - Tag for teams:
 1. Ex: #demand-generation, #product
 - Tag for okr:
 1. Tag for Quarter: ex: Q4FY19
- d. Post the Idea link in #team-slack-channel and schedule a time to present at *Experiment Review Meeting* by pinging {Program Manager}.

2. Implement (Idea Stages: “Development”, “Setup/QA”, State: “Active”)

- a. Create an experiment in Optimizely and “link” it to your Program Management Idea (or add to **experiment roadmap spreadsheet**)
 - If you need support building an experiment, please {**follow these steps to get technical support for experiment build**}.
- b. Add the URL for your idea in Program Management (or **experiment roadmap**) in the description of the experiment.
- c. Work with the appropriate engineering owner to build out the experiment in their area of the code base or build yourself following experimentation best practices {link to experiment code implementation and conventions documentation}
 - If you need to learn more about Optimizely Full Stack, developer documentation is [here](#).
- d. Full Stack experiments must undergo a code review similar to any other code being deployed into our code base before going live, {link to code review process}.
- e. Any experiment which changes the UI requires a design review before it goes live. If you need help with design to prepare for experiment review, submit a request {**link to request process**}.

3. Run (Idea Stage: “Testing”)

- a. Please communicate about your experiment launch at least 24 hours in advance:
 - Add a comment to your Idea giving followers a heads up when the experiment is scheduled to start
 - If your experiment will be visible to customers, send an email to **team@yourcompany.com** detailing your hypothesis and variations
 1. Visible Changes template email: {**link to template**}

4. Analyze (Idea Stage: “Analysis”)

- a. Schedule a slot in Experiment Review Meeting to go through your experiment results
- b. Record your results in Program Management (Record Results & Add Notes) (or **experiment backlog**) and share with the team via **team@yourcompany.com**
 - Visible Changes Results Sharing template email: {**link to template**}

5. Rollout

- a. Depending on the experiment and results, this may include moving 100% of traffic to the winning variation, removing experiment code from your codebase. You should discuss with your team ahead of time and be prepared to walk through your rollout plan during Experiment Review!
 - If you are leaving your experiment as a hotfix for a certain amount of time, log a JIRA ticket with the engineering team that would need to incorporate it and label the experiment in Optimizely with [Hotfix]
- b. When you are ready to rollout, follow the **instructions for launching a rollout [here]**

6. Archive (Idea Stage: “Complete”, Status: “Archive”)

- a. Before you archive the experiment (or feature), make sure to implement in code the behavior you need.

7. When archived, the datafiles assume that no such experiment or feature exists and so any reference to the experiment or feature will result in the default behavior. So make sure to update the default behavior before archiving.

Additional Help

Where can I learn more about how to implement my experiment?

See the Optimizely's Full Stack documentation here: <https://docs.developers.optimizely.com/full-stack/docs>

What are our naming conventions?

- Experiment Name: [Owner First Name] Touchpoint Name - Idea Name
Examples:
 - [Whelan] Page Builder - Dynamic Websites Awareness (v2)
 - [Zach] Navigation - Program Management Plug
- Description: Link to Program Management Idea or other relevant documentation
 - If a hotfix, include JIRA link for hotfix implementation into codebase
- Variation Name: Description of Variation (please do not use default variation names like "Variation #1"--but do keep it short and descriptive.

What is the weekly Experiment Review meeting for?

We'll use Experiment Reviews to go through experiment plans and results together and provide feedback and advice to the teams running them. The goals of this meeting are to 1) ensure we run high-integrity experiments and campaigns and 2) spread knowledge and best practices around our team.

I have questions. Where do I go?

Please add questions directly to this document, or post them in **#experiment-team-channel** or ping {program manager}.

Creating New Entities for Optimizely Web



{Shortlink for this documentation}

Welcome to the manual for creating and updating Optimizely X Web entities at **{your company}**. It's intended to provide guidance to anyone working on an Optimizely Web project.

Optimizely X Web Entity is the general name we use for Pages, Audiences, Extensions, or any of the several other re-usable entities in Optimizely X Web. This manual is for the **{these websites you experiment on}** and should be used as the canonical guide to building Optimizely X Web entities.

Using this Manual

This manual will cover canonical guidelines for the following Optimizely X Web entity areas:

- Name
- Description (if applicable)
- Configuration inputs / options (if applicable)
- Custom JS guidelines (technical) (if applicable)

The goal of this manual is to ensure above all else we **do not degrade app performance** and that we configure Optimizely X Web entities to be as reusable as possible. Review is needed for entity creations and updates that contain Custom JavaScript. To schedule an entity review, create **JIRA** ticket and assign to **{Program Manager}**. **{Program Manager}** will coordinate a review with a **{code reviewer}** from there. Please expect 1-2 business days for the review to be assigned - the review process will vary on review size, scope, and feedback.

IMPORTANT: the name prefix “[c]” stands for canonical and should only be added by the Program Manager if the Optimizely X Web entity has been reviewed and approved.

Table of Contents

Two asterisks (**) indicate that the changes to this entity type go live as soon as they are saved, because it is included in snippets regardless of it being used. External review of the entity config is necessary before actually creating this entity. Refer to the specific section on how that review should be coordinated.

Creating and Updating Optimizely X Web Entities

- Using this Manual

- Table of Contents

- Using ES6 JavaScript

- Using the {internal data layer}

- Entity Types

 - Audiences

 - Exclusion Groups

 - Extensions

 - Custom Analytics Integrations

 - Custom Events

 - Pages / Pageview Events / Click Events / Visual Tags

 - Pages

 - Pageview Events

 - Click Events

 - Visual Tags

- Project JS **

 - JIRA Template

- Projects

- Teams

- Custom Snippets

Using ES6 JavaScript

Any considerations around JavaScript versions?

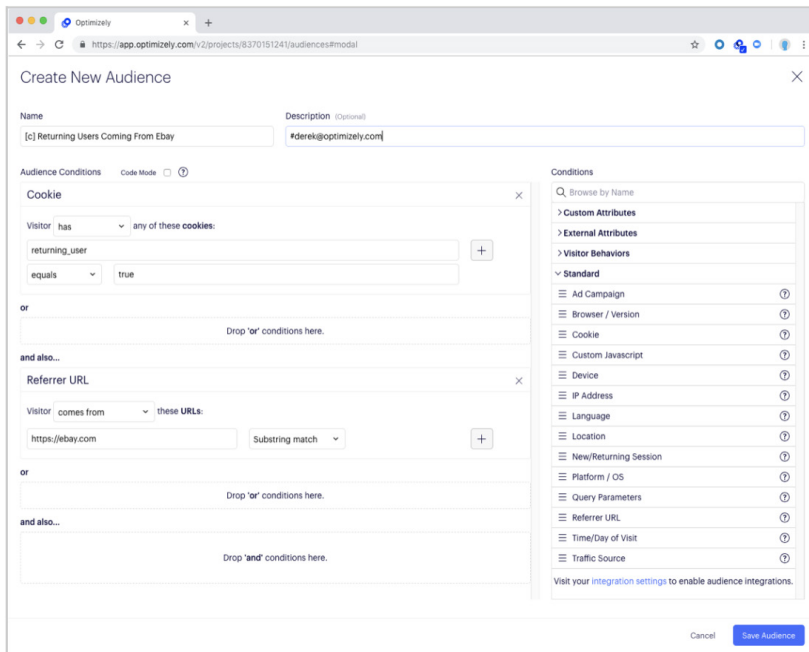
Using the {internal data layer}

Do you have a data layer that can be leveraged?

Entity Types

Audiences

This entity type has historically been one of the most susceptible to needless duplication. We should be creating high quality Audiences that can be reused and combined with other audiences. Pages should be thought of as the where, and Audiences are the who.



Name

Short description of the who. Example:
[c] First Time Users

Description

Be very specific here. In our example above with the [c] First Time Users Audience, go into detail on what “first time” really means. Additionally, add a hashtag (#) with the creators’ email prefix at the end. Example:
When a user first signs in to the Optimizely app, they have a “first_visit” session cookie added. If that cookie is present, this audience will be met. #derek #becca

Configuration inputs / options

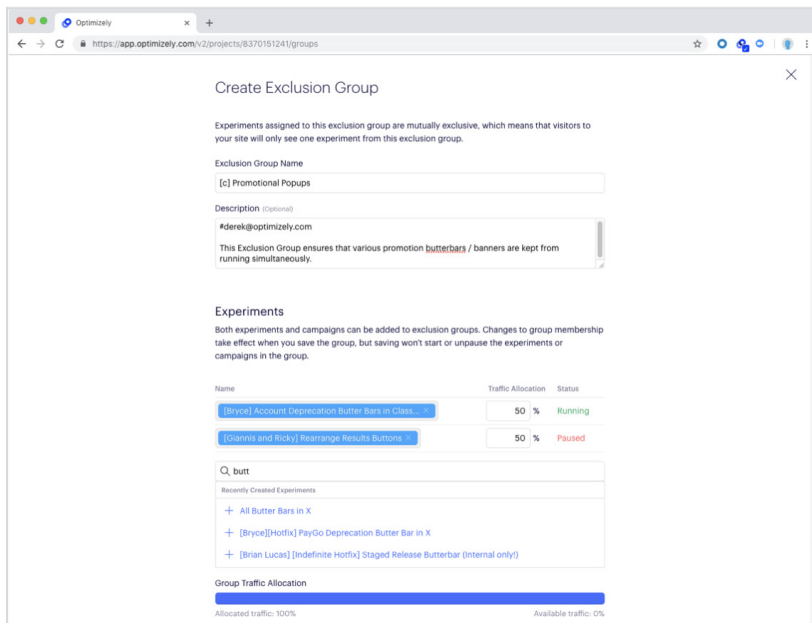
Audiences are made up of conditions, please refer to [Audience conditions: Descriptions and examples](#) on the Optiverse for more info.

Custom JS guidelines (technical)

In Optimizely X Web, there is an Audience condition type called Custom JS. Custom JS in Audiences, just like anywhere else, should only be used when absolutely necessary. When Audience Evaluation occurs (learn about [How the Optimizely X snippet works: Order of activation](#) on Optiverse), the saved Custom JS is immediately evaluated as a JavaScript expression. If the expression is truthy, the Audience passes, if falsy or an Error is thrown, the Audience fails. This means only synchronous JavaScript may be used within this Audience condition (e.g. **none of the following may be used:** polling, setTimeout, ES6 Promises, etc.). If the info you're trying to access from within the Custom JS Audience condition may not be available, you should leverage the upstream *Pages* entity to delay Audience evaluation.

Exclusion Groups

Exclusion Groups can be used to create mutually exclusive groups of Experiments and Campaigns.



Name

Short description of what this Exclusion Group is keeping mutually exclusive. Example: [c] Promotional Pop-ups

Description

Be more specific here. In our example above with the [c] Promotional Pop-ups Exclusion Group, go into detail on why an Exclusion Group is needed. Additionally, add a hashtag (#) with the creators' email prefix at the end.

Example: {Company} has a few different popup campaigns running at any given moment. In order to avoid the over exposure and frustration of the user, only one should be shown at random and the others prevented from being activated. #derek

Configuration inputs / options

Exclusion Group configuration is pretty straightforward. Please refer to [Mutually exclusive experiments in Optimizely X Web](#) on the Optiverse for more info.

Custom JS guidelines (technical)

N/A

Template content trimmed, download for full template





Section 3

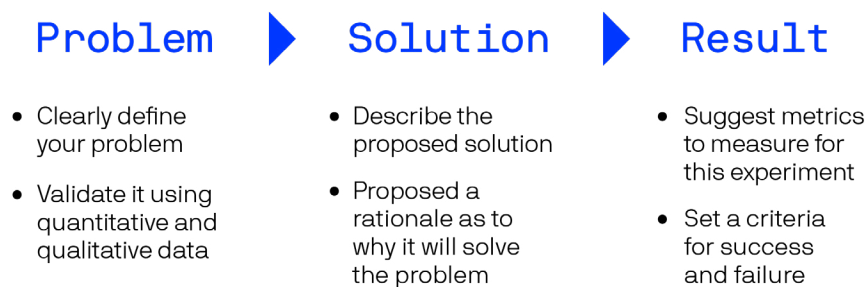
How to get Experiment Ideas from Everywhere in your Organization & Creating your Roadmap

It is important to generate a steady stream of ideas from across the organization to ensure you have a full backlog to pull from, keeping you from being flat-footed around what experiments to prioritize and build next. A running list of experiment ideas that everyone can contribute to encourages more people to participate in experimentation, which is good for your program, your product, and your company. Every person in your organization has a unique perspective, based on their own utilization of the product or interactions with customers, that can provide great, new ideas to explore.

So, how do you get all these ideas? A few tactics I've seen work well:

- **Passive/Scalable:**
 - Set up and socialize a physical box for adding in experiment ideas
 - Set up and socialize a Google form for submitting experiment ideas
 - Add everyone to your Program Management instance as a Teammate so they can submit ideas directly
 - Socialize your roadmap spreadsheet and give everyone comment access
 - Designate a Slack channel for submitting ideas
- **Active/Less Scalable:**
 - Run Ideation Sessions with each Team
 - Sit with individuals to understand what they're working on for the quarter and how experimentation can impact their key metrics and/or launches
 - Do [User Journey Mapping](#)

At Optimizely, running an ideation session with a team is a tactic I've seen be especially helpful to get a large amount of experiment ideas in a short period of time. My colleague [Alek](#) wrote a great [blog post](#) outlining step-by-step how to run a session with your design team, which goes through the entire process for facilitation, following this format:




I run ideation sessions by bringing together a specific team and have them focus on one specific page on the website. The important piece is to pick something that the team will have a lot of opinions on and areas they want to experiment on to improve.

Once you are ready to have an ideation session, the templates below will help you to capture:


- All the ideas from your ideation session and organize them by theme
- A framework for scoring your ideas
- An Experimentation Roadmap template that holds all your ideas from the session and beyond. You can also leverage [Program Management](#), as it has great sorting of ideas by both product area and type, has scoring for streamlining prioritization, commenting built-in and connects to your experiments in Optimizely.

Experiment Roadmap


 Download Template

 Experiment Roadmap Template

Prioritize							Status	Hypothesis Problem <small>(with validation)</small>
Experiment Name	Potential 1-5 <small>5= most likely to win</small>	Business Impact: 1-5 <small>5=highest ROI</small>	Technical Effort: 1-5 <small>5 = least challenging</small>	Love: 1-5 <small>5 = LOVE</small>	Prioritization Index <small>(Average)</small>	Priority <small>(High, Medium, Low)</small>		
Free Trial	3	3	1	4	2.75	HIGH	Idea	
Call CTA on Cart Page	1	3	4	2	2.5			

 Experiment Roadmap Template

Describe						
Hypothesis Solution <small>(with rationale)</small>	Hypothesis Result	Description	Pages	Audiences	Start Date	End Date

 Experiment Roadmap Template

Find Out How Long To Run Tests For					Minimum Lift You Can Detect After...				Enter Your Results			
Primary Goal Name	Baseline %	Total Variations	Daily Unique Visitors	Traffic Allocation	5	7	2	5	Result % <small>(Lift/Loss)</small>	Implemented <small>(Y/N)</small>	Why did the experiment...	What should we...
					Days	Days	Weeks	Weeks				
				100%								
				100%								
				100%								
				100%								
				100%								

Ideation Session Problem & Solution Mapping

 [Download Template](#)



Ideation Session Problem, Solutions and Themes Template

Problem	Problem Theme	Sub Theme




Ideation Session Problem, and Themes Template

Problem Theme	Solution	MGMT Ideas	Result	Product


Experiment Idea Scoring Framework

 Download Template

 Scoring and Prioritization Framework

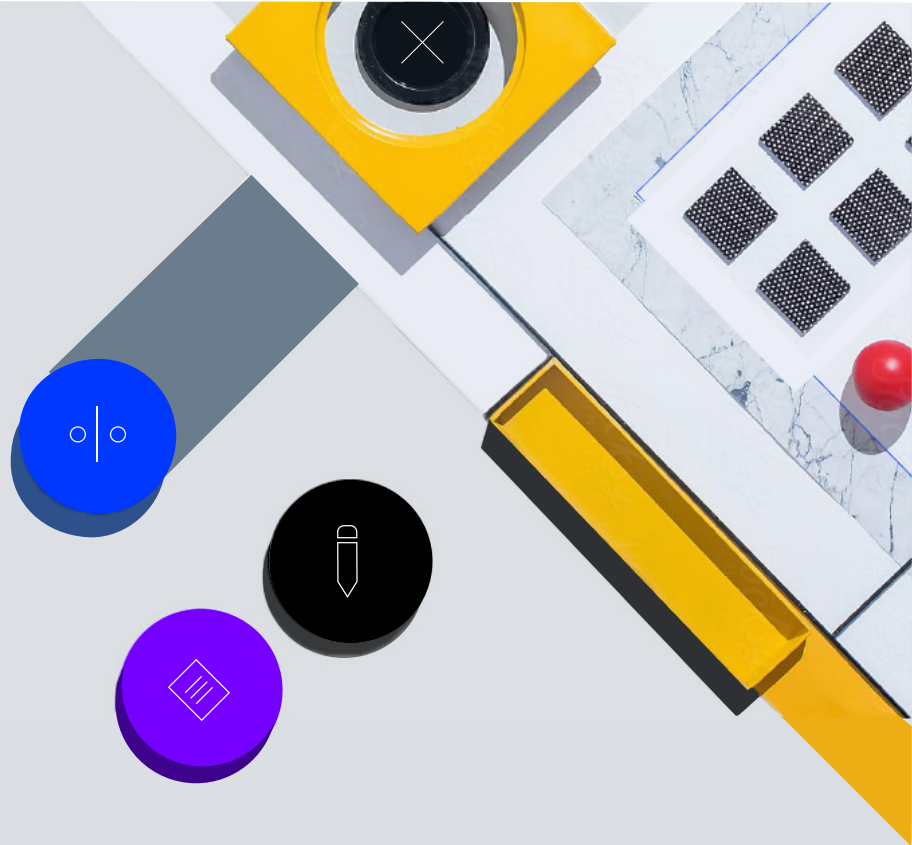
Prioritize Use this tab to prioritize ideas to test after a brainstorming session, after sourcing ideas internally at the company, or after doing detailed research on your own or with your team about your digital experiences.

Experiment Idea	Priority (High, Medium, Low)	Business Impact: 1-5 (5+ highest ROI)	Technical Effort: 1-5 (5+ most challenging)	Prioritization Index: Sum of Scores
	High	3	2	1
				0
				0
				0
				0

 Scoring and Prioritization Framework

Prioritize Use this tab to prioritize ideas to test after a brainstorming session, after sourcing ideas internally at the company, or after doing detailed research on your own or with your team about your digital experiences.

Experiment Idea	Score	Main Metric	Location	Fold	Targeting	New Information	Benchmarking	Strategic Topic	Mobile
	4	1	0	1	0	0	1	1	0
	0								
	0								
	0								
	0								



Section 4

Reviewing and Prioritizing Experiments

A framework for prioritization is especially helpful when you have a high velocity of experiments. It is also useful when you have a high number of experiments to run in the same area of the site as you want to ensure transparency across the organization for how these trade-offs are being made.

Arriving at this point is a good thing! It means your team has a ton of ideas and enthusiasm for experimentation. This point often happens sooner if you have a long lead time for launching an experiment from build, QA, and launch phases. To allow you to move quickly to a structured prioritization framework, I recommend getting your team into the habit of scoring experiments early so you build the muscle of prioritization early, before you need it.

A scoring and voting structure can help with democratizing your process. It allows everyone to voice their perspective on what would make the biggest impact. If you're using Optimizely's Program Management, scoring is built directly into the system. If you're using a spreadsheet instead, I recommend creating an experiment roadmap that includes scoring columns next to each experiment idea, which is already built-in to the experiment roadmap spreadsheet template included in this toolkit.

To support Optimizely's experiment prioritization needs, I run a weekly meeting called Experiment Review to ensure experiments are ready to be launched from a hypothesis, metrics, and set up perspective.

For Optimizely, Experiment Review serves multiple functions:

- Lightweight gating and prioritization for experiments
- Ensuring a high quality bar is met with experiments being launched
- Up-leveling hypothesis thinking and experiment set-up across the organization
- Visibility for what is about to be launched

To help you launch your own Experiment Review meeting, I've included a template for running the meeting and agenda. I've also included an Experiment Brief template that can be used to ensure experimenters come to the meeting with all the relevant information needed to discuss their experiment idea.

Experiment Review Meeting: Agenda, Notes and Backlog



{Shortlink for this documentation}

A short description of what Experiment Review is and how it is used to help support the team.
(example below)

- Why is this meeting impactful for our program and company?
- What we do and don't discuss?
- When you have to come to this meeting?

To request an upcoming slot, please post in {SLACK CHANNEL NAME or PERSON TO MESSAGE}

Example Presentations

Examples of good presentations that have been given historically can provide helpful frameworks to newer experimenters. Your team can also present directly from inside the Optimizely app or Program Management.

- **Experiment Review:** {link to presentation}
- **Results Review:** {link to presentation}

On Deck

What Experiments or Results reviews are in your backlog?

- [Name of Owner/Presenter] Name of Experiment Idea with link to experiment or idea in Optimizely

{Current Quarter}

This is where you can keep a list of all Experiments your key stakeholders have committed to for the quarter, at least as possibilities.

At Optimizely, this is all the ideas that Product Managers or Marketing Team are considering for the Quarter.

- Name of Experiment Idea with it linked

{Date of the next next meeting}

This is where you can keep a list of all Experiments that will happen in the next next meeting. Seeding this as early as possible will help with the meeting having content each week.

- [Name of Owner/Presenter] Name of Experiment Idea with link to experiment or idea in Optimizely

{Date of next meeting}

This is where you can keep a list of all Experiments that will happen in the next meeting. Should be sent via your instant messaging platform to inform everyone of what will be discussed (example below)

- [Name of Owner/Presenter] Name of Experiment Idea with link to experiment or idea in Optimizely

Example of these notes:

go/experimentreview

Weekly ADEPT Experiment Review

We'll use experiment reviews to go through experiment plans and results as a team and provide feedback and advice to the teams running them. The goals of this meeting are to 1) ensure we run high-integrity experiments and 2) spread knowledge and best practices around our team. This is an approval meeting but not a prioritization meeting.

To request an upcoming slot, please post in #experiment-review.

After Experiment review and If your experiment contains custom JavaScript, follow the instructions in the [Optimizely on Optimizely Web Experimentation Review](#) docs.

Example Presentations

- **Results Review:** [\[here\]](#)

Q1


- **Experimentation + Documentation**
 - [Add Dev Docs into App](#)
 - [Verticalized audience promotion for onboarded users](#)
 - [Tip of the Week](#)
 - [\[Pendo\]\[Evergreen\]\[Hotfix\] MVT Section Rollups in-product education](#)

Example for sharing agenda via instant messaging before meeting:


#experiment-review
☆ | 👤 71 | 🗨️ 1 | go/experimentreview

📞 ⓘ ⚙️ 🔍 Search @ ☆ 🎁 4

February 26th, 2019

 **Becca** 2:28 PM
@here agenda for tomorrow's experiment review below 📎
[@travis] Cache UserWhitelist queries that every single http call (API call and page load) executes unnecessarily
[@whelan] Developer Mode
[@whelan] Results Review: Label "No jQuery" as "Recommended"
[@jamie @quintond] Results Review: Full Stack Manager
Look forward to seeing y'all tomorrow!
👍 2 🗨️

March 5th, 2019

 **Becca** 1:01 PM
@here agenda for tomorrow's experiment review below 📎
[@Breana] Drive Extensions Usage
[@giannis] Results Review: Painted door for Flexible Stats
[@jamie @quintond] Results Review: Full Stack Manager
Look forward to seeing y'all tomorrow!

Name of Experiment

Owner(s):

Experiment:

Requested Reviewer	Approval Status

Implementation

General

What is the hypothesis for this experiment? A hypothesis is a prediction you create prior to running an experiment. The common format is: If [cause], then [effect], because [rationale].

Hypothesis

Details about the experiment at a high level

Metrics

1. Metric 1
2. Metric 2

Targeting

1. Where is this running?
2. Who is seeing it?

Variations & Bucketing

Screenshots of the variation designs.

What percentage of visitors will see each variation?

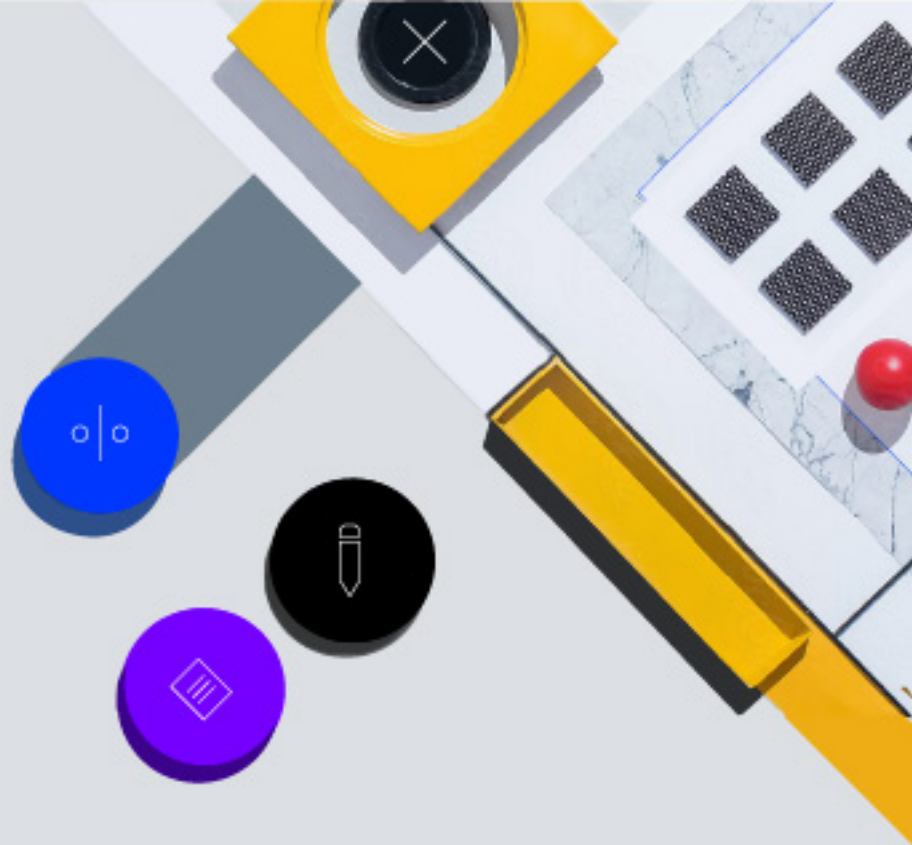
Analysis

What additional baseline data or analysis can you provide for this experiment?

Notes

What feedback or gotchas do you have to share with internal teams?

E.g. any bugs? Process or product gaps?



Section 5

How to make Experimentation Visible Consistently Across your Organization

Program visibility is one of my favorite topics and I'm a believer that it's important to use multiple methods for sharing information. As much as possible, aim to "meet people where they already are," using existing email lists and meetings to piggyback on that audience and process.

For Optimizely on Optimizely, the name for our internal experimentation program at Optimizely, I use a variety of tactics to make the program visible:

- Create physical walls that represent experiments
- Leverage existing meetings or processes to share experimentation updates
- Send company-wide emails when an experiment is launched or completed
- Regularly present interesting learnings from experimentation during team meetings
- Share learnings to specific groups on Slack who would be most interested
- Blog to share new tools, tactics or learnings about the program externally

Below, you'll find examples and templates of the company-wide emails I use for launching experiments, experiment results and pictures of the physical walls representing the experimentation program over the years. These emails are lightweight by design. I recommend sending these in advance of the experiment launching and to as large an email alias as possible. I personally send them company-wide.

At Optimizely, we have an email alias called "Visible Changes" that originally existed for sharing changes to the product with the whole company. Since experimentation also makes changes to the look and behavior of the product, I piggybacked on that email alias and started having experimenters send experiment launch and results emails with the title "[visible-changes][experiment]".

I ask the person who had the idea for experiment send the email, for several reasons:

- If I send the email every time, people are more likely to tune it out
- Seeing teammates contribute to experimentation and getting company-wide visibility can encourage other team members to contribute via the "Culture of FOMO" and wanting to get that sort of visibility as well
- If people have questions, they know who to go to with questions based on who sent the email

When you have insights to share - responding to your launch email with the result is a great way to follow-up on the experiment and close the loop about the impact that experimentation has made. This also heightens and maintains the visibility of the overall program.

Below, I've shared photos of physical walls I built. The first images show all experiments individually by status within agile product delivery. The second set shows a summary of all experiments currently running or planned against quarterly goals. I find creating physical spaces to show experimentation program status is especially helpful when getting a program off the ground as it helps with bringing experimentation out of the digital world and into the physical world where people can see it tangibly where they already are.

In the first picture, these cards live on our [wall of work](#), which represents all engineering work from dreaming to doing. Showing experimentation work represented in this space makes sense as in-product experiments are driven by those teams and prioritized similar to other product work. Additionally, we already have a weekly meeting in place where we share updates on the Wall of Work items. I join to share experimentation updates or, even better, have the product managers themselves share updates by adding these cards to the already existing wall.

Experiment Start Email Template

 Download
Template

To: you@yourcompany.com

Bcc: team@yourcompany.com

Title: [experiment] Name of your Experiment

Provide a brief summary of what this experiment looks like and seeks to impact.

TL;DR: Today, we launched an experiment to {what does this look like} and {why are you making this change}

What are we launching?

- **What?**
 - *What sort of change is happening in this experiment?*
- **Hypothesis?**
 - *What is your experiment hypothesis around the impact this experiment will have?*
- **Where?**
 - *Where is this experiment running?*
- **Who?**
 - *Who is able to see this experiment?*
- **Metrics? What does Success look like?**
 - **Primary Metric:** *What is the main metric you are looking to impact?*
 - **Monitoring Metrics:** *What are the other metrics you are also monitoring and tracking as part of this experiment?*
- **When?**
 - *When is this experiment being launched?*

What does the variation look like?

{Screenshot of Variations}

Where can I learn more?

- **Experiment in App** {link to Experiment in Optimizely}
- **Idea in Program Management** {link to idea in Program Management or other experiment idea tracking system}

Thank You!

Who helped with the build and launch of this experiment?

- {Coworker} for the Idea
- {Coworker} for Experiment Build support
- {Coworker} for Design support
- {Coworker} for Shepherding support
- {Coworker} for code review

If you have questions, please reach out to {Experiment Owner Name} .

Optimizely example:

[visible-changes][experiment] Add "Link Experiment" text link to Idea List in Program Management Changes x



Rebecca Bruggman <rebecca@optimizely.com>
to me, bcc: visible-changes -

TL;DR: On Thursday, we launched an experiment to add a "Link Experiment" text link in Program Management to all of our customers.

What are we launching?

- **What?**
 - Add a text link to each idea on the Program Management home page when an idea is in "Development", "Setup & QA" or "Testing" and nothing is currently linked.
 - Currently, this area is blank space if no Experiment is currently linked and notes an Experiment is linked if one is already linked.
- **Hypothesis?**
 - We don't get what we don't ask for :)
 - If we ask our customers to link Experiments, this will increase the number of Experiments that are linked to ideas in Program Management, which in turn helps with retention.
- **Where?**
 - On Idea List Pages in Program Management.
- **Who?**
 - Everyone with access to Program Management.
- **Metrics? What does Success look like?**
 - **Primary Metrics:** Increase in experiments linked, both total and unique
 - **Monitoring Metrics:** Idea Details Page Viewed and Idea Status Changed
 - Overall, encouraging our customers to do actions that contribute to retention seems like a win, so this is gathering additional data to see how customers react to this update.
- **When?**
 - This is live now!

What does the variation look like?

Owner	Stage	Assignee	Experiments
Brad Rozran Idea List Page In... Add this experiment to your app, then will...	Setup & QA	Rebecca Bruggman Started Jan 11, 2018	Link Experiment

Where can I learn more?

- [Experiment in App](#)
- [Idea in Program Management](#)

Thank You!

- Brad Rozran for the Idea
- Simone Coelho for Experiment Build support
- Shane Fontane for Design support
- Kris McKee for Shepherding support
- David Mann for Code Review

If you have questions, please reach out to Becca (me).

Results Share Email Template

 Download
Template

To: you@yourcompany.com

Bcc: team@yourcompany.com

Title: [experiment] Name of your Experiment

If you sent an email when the experiment started, you can respond to that email with the below information. This helps with providing the original experimentation context with this results update. Provide a brief summary of what this experiment looks like and a summary of the results

TL;DR: We launched an experiment to {what hypothesis are you looking to measure} and {why are you making this change}. After running it for {amount of time}, we saw {what results were observed}

Background

What is the background context for why this experiment was run?

Hypothesis

What is your experiment hypothesis around the impact this experiment will have?

Control

{Screenshot of Control/Original}

Variation

{Screenshot of Variation(s)}

Metrics

How long did the experiment run?

Visitors: {Number of Visitors}

What is a short summary of the metrics below? Did you hit statistical significance?

Primary Metric

- ↓ Expected: What did you expect to happen with this metric?
- ↑ Actual: XX% increase (not stat sig)

Monitoring Metrics

- ↑ Expected: Summary of Metric and what you expect to happen
- ↑ Actual: XX% increase (not stat sig)

- ↓ Expected: Summary of Metric and what you expect to happen
- ↓ Actual: XX% decrease (stat sig)

- ↑ Expected: Summary of Metric and what you expect to happen
- ↓ Actual: XX% decrease (stat sig)

Questions & Observations

- *What are some key takeaways from this experiment?*

Follow-up

What are next steps for you and your team? Rolling this experiment out to 100%? Iterations on this experiment?

Optimizely example:

BACKGROUND

Optimizely's messaging and positioning has changed since we launched the original [developer hub](#). We'd like to refresh this page to reflect how Optimizely enables [developers](#) in 2019 and highlight our most important resources.

HYPOTHESIS

This is a "do no harm" experiment, because we know we want to update the messaging and design. If we create a redirect experiment to do the following ...

... we will see an decrease in **bounce rate** as users find more relevant content.

METRICS

Running for 2 months
Visitors: 2.836

No statistical significance on primary metric. A few secondary metrics have reached stat sig.

Primary Metric

- ↑ Expected: Decrease bounce rate from homepage
- ↑ Actual: 9% increase (not stat sig)

Monitoring Metrics

- ↑ Expected: Increase clicks to important links
- ↑ Actual: 26% increase (not stat sig)
- ↑ Expected: Decrease clicks to search bar
- ↑ Actual: 44% decrease (stat sig)
- ↑ Expected: Increase clicks to all links on page (in retrospect, not a good metric)
- ↑ Actual: 29% decrease (stat sig)
- ↑ Expected: Decrease exit rate from [developer docs](#) pages
- ↑ Actual: 6% decrease (not stat sig)

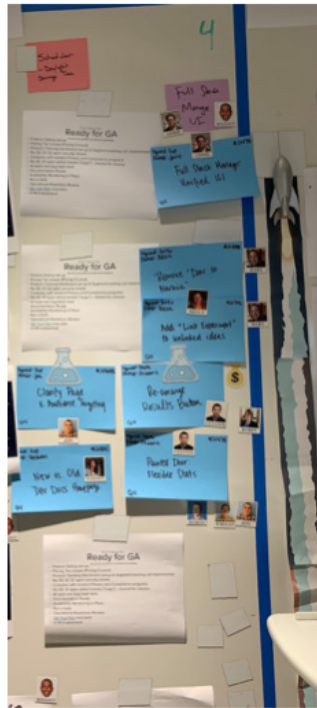
QUESTIONS & OBSERVATIONS

- Why are users bouncing more? It's not stat sig, but if this is an indicator -- is it because they aren't finding content that was removed (OTT/Mobile or the [Developer Account](#) signup form)? If so, should we just add OTT/Mobile to the Full Stack blurb?
- Other trends generally positive. Good signals that the new page is clearer.

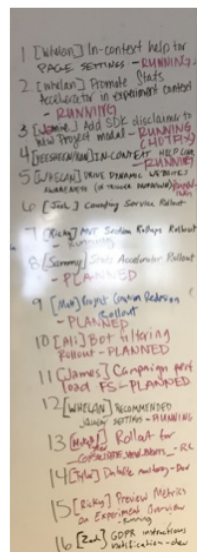
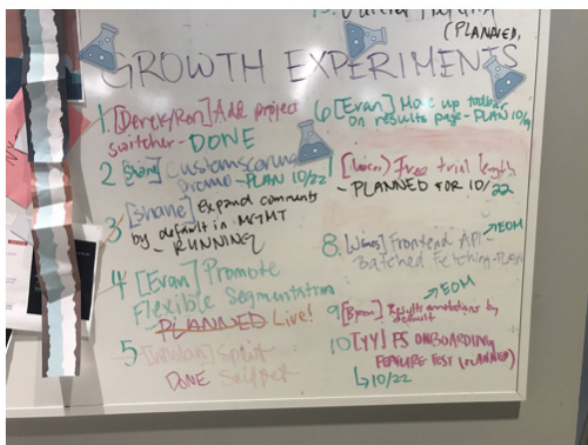
FOLLOW-UP

Quick sync with the Strategy Consultant team to go over results. Discuss in Experiment Review.

Visualizing Experimentation Program Examples



Examples of including experimentation into product delivery agile processes



Examples summarizing all current experiments running and planned against a specific quarterly goal



Section 6

Demonstrating ROI

You can make a huge impact for your program by sending or presenting a summarized, executive-level digestible recap. A digest helps you demonstrate the overall ROI of your experimentation program. Tie it back to top line business goals, and show the additional value of the program, such as risk-mitigation, acceleration of customer learnings or cost savings by not launching an experience to 100%.

I recommend sharing your recaps at a consistent cadence. At Optimizely, we do quarterly goal setting of Objectives & Key Results (OKRs). I send an email update of the program against our OKR goals to align with that cadence and piggybacking on processes that already exist.

Below you'll find a template for sharing a summarized, program-level email update to your whole company and a slide deck for presenting across your organization.

When preparing your recap email, it can be helpful to think through the ROI of your program as it aligns to top level metrics for your company. This usually entails assigning a value to specific user actions and then identifying how much experimentation has influenced these behaviors. If you have your goal tree created from a few sections back, you can assign values to those events to help with your ROI calculation. If you're getting started or looking for some additional guidance on how to approach ROI for experimentation, Optimizely has a [great ROI calculator](#).

I include a summary with the total number of experiments we've run, any scalable work such as blog posts or trainings that have happened, a plug for experiment review, make sure to thank the rest of my team and a table that has the top level information for all experiments launched. Finally, I include all the details from the experiment launch emails below so it's all in one summarized place. If you've been leveraging the Experiment Start and Experiment Results emails from the previous section, creating a summarized recap is very streamlined.

I know that making a program visible can feel intimidating. But if you don't make the work visible, no one will know it's happening and it will be difficult to ask for support and resources when you do need help.

Sometimes, you even get a congratulatory email from your CEO - which can make it all worth it!

Experimentation Center of Excellence Update - Q4FY19 Changes x



Rebecca Bruggman

Hi Optinauts, Happy FY20! We had a great fourth quarter for Experimentation on ADEPT and hit all our OKRs with 9 €

9



Ashley Law

Sounds good, thanks!



Jay Larson

to Claire, Carl, Kate, Jennifer, me ▾

Hi Becca,

There's some great content here. I'm curious: How does this content get shared with our customers?

Love this, want to make sure we get this in front of our customers as well as prospects.

Please let me know.

Thanks,

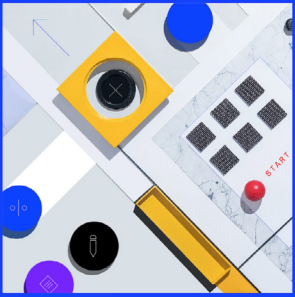
Jay

Experimentation Program Quarterly Update Presentation

Download
Template

Experimentation Recap

Your Name



Experimentation Program - X Experiments (OKR: X)

Experiment Name	Squad	Details	Primary Metric	Links

Team that Ran Experiment: Name of Experiment

Screenshot of Experiment

- Experiment details
- More Experiment details


Goals for next quarter

Experiment Goals



- X Total Experiments
 - X for this team
 - X for this team
 - X for this team

Another Experimentation goal - more visibility?


Example →



Thank You!

My amazing Account Team,
Gabby and Alek!



Fantastic documentation!

Everyone who participated in Experiment Review, Submitted an Idea or launched an experiment -- thank you!

Quarterly Experiment Program Update Email

↓  Download Template



Hi Team,

{Greeting}! We had a great {time period} for Experimentation and {update on goals and whether y'all hit them}.

Any process updates to share? Updates to documentation? Blog posts or conference talks to share?

Call out ways your team can get involved in experimentation, such as Slack channels they can join or meetings to attend.

Thank people for contributing to experimentation or anyone who went above & beyond.

Happy Testing!

Your Name

Experimentation at {Company Name} {Time Period}

In {time period} we launched XX experiments and XX rollouts, including {types of experiments the team ran this quarter}. I've listed all Experiments and Rollouts below and I've also pulled out a few experiments of note from the quarter with additional details from the Experimenters.

Experiment Name	Team	Details	Experiment Status	Links
<i>Name of Experiment</i>	<i>Name of Team</i>	<i>High level details for Experiment</i>	<i>Running Paused, not Statistically Significant Paused, Statistically Significant</i>	<i>Idea in Program Management {link to idea in Program Management or other experiment idea tracking system}</i>
<i>Name of Experiment</i>	<i>Name of Team</i>	<i>High level details for Experiment</i>	<i>Running Paused, not Statistically Significant Paused, Statistically Significant</i>	<i>Idea in Program Management {link to idea in Program Management or other experiment idea tracking system}</i>

Rollouts

Name of Rollout	Team
<i>Name of Rollout</i>	<i>Name of Team</i>
<i>Name of Rollout</i>	<i>Name of Team</i>
<i>Name of Rollout</i>	<i>Name of Team</i>

Repeat this template below for as many experiments as you'd like to share.

{Web/Full Stack} Experiment: {Name of Experiment}

Owner: {Name of Experiment Owner}

Idea in Program Management {link to idea in Program Management or other experiment idea tracking system}

{Screenshot of Original and Variations}

Background

What is the background context for why this experiment was run?

Hypothesis

What is your experiment hypothesis around the impact this experiment will have?

Metrics

How long did the experiment run?

Visitors: {Number of Visitors}

What is a short summary of the metrics below? Did you hit statistical significance?

Primary Metric

↓ Expected: What did you expect to happen with this metric?

↑ Actual: XX% increase (not stat sig)

Monitoring Metrics

↑ Expected: Summary of Metric and what you expect to happen

↑ Actual: XX% increase (not stat sig)

↓ Expected: Summary of Metric and what you expect to happen

↓ Actual: XX% decrease (stat sig)

↑ Expected: Summary of Metric and what you expect to happen

↓ Actual: XX% decrease (stat sig)

Questions & Observations

- What are some key takeaways from this experiment?

Follow-up

What are next steps for you and your team? Rolling this experiment out to 100%? Iterations on this experiment?

Optimizely Example:

Experimentation Center of Excellence Update - Q4FY19 Changes x



Rebecca Bruggman <rebecca@optimizely.com>
to Team ▾

Thu, Feb 14, 2019, 12:30 PM ☆ ↶ ⋮

Hi Optinauts,

Happy FY20! We had a great fourth quarter for Experimentation on ADEPT and hit all our OKRs with **9 experiments (OKR: 6) and 100% of Features were Rollouts, 6 in total.**

This quarter, we also created great new content you can share with your customers:

- [Optimizely Under the Hood Webinar Series: Managing Experimentation at Scale](#)
- [How to set your goals to level up your Experimentation Program](#)
- [Experimentation Reading List: 5 ways to level up your experimentation skills right now](#)

In January, I sat down to clean up our Optimizely App web project to ensure that we only have Audiences, Pages and Metrics that we are actively using moving forward. This effort reduced the number of components in our web project by over 60%.

To ensure we have a process moving forward for creating new Audiences, Pages and Metrics, Derek Hammond put together an amazing [Product Requirements Document](#) on how to create these new components from code reviews to nomenclature.

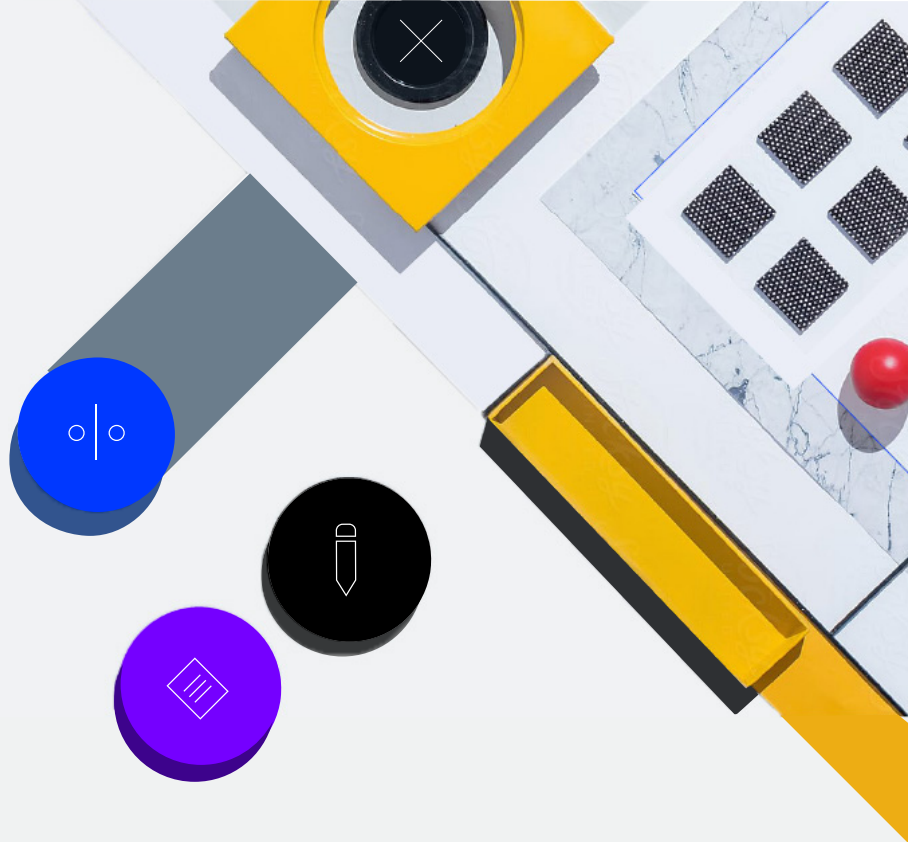
Join us at [Experiment Review](#) every Wednesday at 10am in Pioneer and on Slack at [#experiment-review](#).

Interested in seeing how we run the Experimentation Program on Product? We review all experiments and experiment results that go into the product with discussions often getting quite lively. This meeting is open to everyone and a great way to get insight into how our customers run their own programs.

A huge thank you to everyone who joined us at Experiment Review, submitted an idea to Program Management or launched an experiment this quarter!

Finally, thank you to my amazing account team, Gabby Des Georges and Alek Toumert.

Happy Testing!
Becca



I hope you enjoyed the Experimentation Program Toolkit!

If you have any feedback or follow-up questions, please tweet me [@bexcitement](#).

If you're looking for additional resources, a few you may find helpful:

→ [Optimizely Program Management](#)

Learn more about Optimizely's Program Management capabilities

→ [Optimizely Rollouts](#)

Free feature flags to support safely rolling out new features to your customers

→ [Alek Toumert's blogs](#) on scaling experimentation

→ [Brian Balfour's blog](#) on maximizing learning with experimentation

Happy Experimenting!

Rebecca Bruggman