Experiment Value Framework

EXPERIMENTATION

Solution	**Experiment**
Use Case	**Feature Experimentation / Omnichannel Experimentation**
Products	**Feature Experimentation**
Summary	Optimizely Feature Experimentation (FX) is designed to help product and engineering teams build more impactful features by combining feature flagging, feature delivery, and experimentation into a single, easy workflow. With its flags-first approach, FX enables product teams to scale no-code experimentation and optimization on top of feature flags without having to scale development. FX helps engineers decouple releases from deployments, enabling engineers to merge and deploy code according to their own guidelines, whilst enabling their Product teams to configure features, changing their configurations through experiments and targeted deliveries according to their own timelines. With a simple feature flag in blast, engineers can easily turn off buggy features and limit their blast radius. It also supports canary testing, where appropriate features can be tested in production, reducing QA effort. All of these features remove friction throughout the Software Development Lifecycle (SDLC). FX helps product teams test features and identify winning configurations for their customers without engineering involvement after initial implementation. FX removes the guesswork of interpreting experiment results without sacrificing statistical rigor. With FX, product teams can be confident about their findings without needing to involve analysts that can slow experimentation velocity.
Before State *Key challenges and problem statements*	**Product Management** - Difficult to understand if and how new features are moving the needle - Difficult to drive a data-informed product development process - Features are rolled out without assessment of impact - Features are built to completion and then discovered as having no impact - If coming from existing solution: - No possibility to test across touchpoints makes it difficult to run comprehensive tests - Every A/B test needs developers. Development resources are scarce and don't see the value of building experiments - Trade-offs between building experiments and building new features - Difficult process / high cost to set up experiments - Low experimentation velocity (number of experiments run) leading to limited value from existing program - Experiment analysis is cumbersome, costly and difficult - It's difficult to interpret the results of an A/B test and making data-driven decisions based on those results - Reliance on Data Science to evaluate A/B tests **Engineering / Development** - Developers are spending too much time building A/B tests - Performance impact of existing A/B testing solution is not acceptable - Maintaining the current experimentation platform is costly - Marketing is making changes to the website that causes errors

	• Difficult to understand impact of features on key engineering metrics like performance, error rates, … • Difficult to implement experimentation in a microservices environment **Data Science** • Too much manual effort spent on analyzing A/B tests • Statistics of existing solution are not trustworthy • PMs are not following good statistical practices evaluating A/B tests (peeking, …)
Neg. Consequences *The result of the "before"*	**Product Management** • Low experimentation velocity (number of experiments) • Development resources wasted building capabilities that don't move the needle • Inconsistencies in data pipelines due to different metrics across many channels • Maintaining development workflows for experiments require lots of overhead (managing the front-end with the back-end) • **Engineering** • Lower dev productivity as measured on building new capabilities • Less dev time spent on new features • Engineering burnt out from boring activities – like changing constants or content • Experiments split across tools (e.g. some in homegrown, some in POS) leading to distrust in results **Data Science** • Work too much on evaluating experiment results vs. building high impact data science products
Future State *Paint a picture* *of the ideal way of working*	**Engineering** • Engineering deploys code behind a feature flag and implements key feature variables • Engineering can deploy code independent of feature rollout whenever they are ready • Engineering can roll back deploys that cause issues quickly and without sleepless nights **Product Management** • Product Managers can run no-code experiments through feature toggles and variables implement by engineers. No additional dev-work required. • PMs learn fast and find the right feature configuration • Consistent workflow (how experiments are deployed and results are read) **Data Science** • Data science is only pulled in for high-impact experiments and define standards for PMs and engineers to follow
Positive Outcomes	**Product Management** • Higher impact of features built • Faster decisions being made

High-level benefits	- More autonomy when running experiments and optimization campaigns **Engineering:** - Less production isses - Faster resolution times - More and faster deploys - Lower blast radius of bugs
Metrics *Business objectives tied to this initiative*	- B2B - No. of leads - Quality of leads - Lead conversion rate - Newsletter Subscriptions … - eCommerce - Revenue - Average Order Value - Conversion Rate … - Media - Pages / Session - Ad Impressions - Subscription conversion rate … - B2C Software (Streaming, …) - Subscription Conversion **KPI – by channel:** ability to compare conversions by location and context **Improved Velocity and Developer Efficiency:** a single UI, data model, and workflow across all channels will allow developers to easily and efficiently deploy any type of experiment across any environment, without needing to learn additional tools or switching between them **Improved Velocity / Scale:** by being able to deploy any type of experiment (front-end and back-end) across many programming languages, experimenters are able to think more broadly across the user experience and target more areas of the funnel with more complex use cases. **Reduced Technical Debt:** because Optimizely is available in four deployments (Web, Server Side, EDGE, and Decision Agent) and 12 programming SDK's, companies can easily adopt new services and programming languages without fearing that it will break their experiment / data stack. Companies can be more agile in reacting to technology trends, because Optimizely is perpetually enabling any new technologies they desire **Improved Retention through better User Experience:** companies can ensure a seamless user experience, where experiments and personalization campaigns remain in sync across devices and channels. This will result in a higher percentage of users returning, and being monetizable in the future - Reach: percent of customers who are part of an experiment
Required Capabilities *Core products, enhancements & features that*	- No-Code Rules Engine - No-Code Experiments on top of Feature Flags - Multiple Concurrent Experiments per Feature Flag - Feature Variations - Multiple targeted rollouts per Feature Flag - Stats Engine (Sequential Statistics)

enable the "After"	• Stats Accelerator • In-Memory Decisioning without performance impact • Optimizely Agent

Differentiators
 	
What uniquely positions us against the competition "How we do it Better"
	Product
	
	• No-Code Rules Engine
	    ○ Multiple Concurrent Experiments per Feature Flag
	    ○ Multiple targeted deliveries per Feature Flag
	• Powerful segmentation and targeting (ODP)
	• CDP Integrations to target features and rollouts
	• Stats Engine
	• Multi-Armed Bandits
	• Unlimited metrics tracked / experiment
	• Fully-featured REST API
	
	Services
	
	• Onboarding and expert services (technical and strategic)
	•

Target Audience *Relevant buyer & user personas by title/seniority*	**Buyer** • Product Leadership • Data Science / Data Leadership • Sometimes: Engineering Leadership **User** • Product Management • Engineering • Data Science
Proof Points *Customer Stories,* *AR & Data Points (e.g. # of experiments run/year)*	[Blue Apron](#) • 10x increase in experiments run per month after adopting Full Stack • 100% number of product teams using experimentation • 37% decrease in customer issues related to unexpected orders "With Optimizely Full Stack, Blue Apron gets statistically valid experimental results on their server and mobile applications which helps them make faster, more informed business decisions. With Optimizely's real-time results, Blue Apron was able to avoid launching a redesigned reactivation flow, which would have resulted in a significant revenue loss. Blue Apron can now quickly and easily build and set up new tests that were simply not possible before." Ambassador Theatre Group (unreleased case study – **INTERNAL USE ONLY**) • 2 experiments run weekly • More than 10% increase in overall checkout conversion "Feature Flags enable ATG to canary test with quicker cycles in a real-world setting without making disruptive code changes. ATG can also quickly roll back tests by a

	toggle of a button when it is set up with a Feature Flag, easing the work of developers and engineers."
Directional Discovery *Leading questions that tie pain to capability*	"How are you measuring the impact of your product releases?" "Where do you run experiments today?" "How many experiments are you running today?" "How much time does engineering spend on creating experiments?" "How long does it take for a full experiment lifecycle?

Solution	**Experiment**
Use Case	**Web A/B Testing**
Products	**Web Experimentation**
Summary	With Web Experimentation, A/B test and experiment at scale. Our proven A/B testing combined with automated personalization and comprehensive experimentation based on real-time customer engagement will boost both engagement and core website KPIs. Learn what experiences customers best respond to whilst building an individual profile of each visitor and recommending the right, personalised information every time. The combination of these learnings drives AI automation for personalization throughout the journey, across applications and on every channel. With Web Experimentation, you will lift your online and application metrics without changing your platform. Web Experimentation ensures the format of your experiences are engaging to visitors by using experimentation, customer data and recommendations across your website and other applications. Drive impactful changes to your website based on tests and customer data with the world's fastest experimentation platform and A/B testing solution.
Before State *Key challenges and problem statements*	**Web Development** - Developers are spending too much time building A/B tests. Current A/B testing platform is not developer friendly. - Performance impact of existing A/B testing solution is not acceptable - Current A/B testing platform is cumbersome to work with and makes writing AB tests difficult **Marketing / Growth PM** - Difficult to understand what changes will drive user conversion - Marketing activities are unguided and changes to the website are driven by management (HIPPO) - Website is not performing conversion wise - If coming from existing solution: - A/B tests are difficult to set up with existing solution - Experiment results are difficult to understand and can't be trusted - Limited ability to set up simple AB tests themselves. Overly reliant on developers - Limited experimentation velocity (no. of tests run) **Analytics**

	• Too much manual effort spent on analyzing A/B tests • Statistics of existing solution are not trustworthy • Marketers are not following good statistical practices evaluating A/B tests (peeking, …)
Neg. Consequences *The result of the "before"*	• Rolling out changes to websites is a gamble and might lead to revenue loss • Unguided optimization activities • Time wasted on website changes that don't lead to improvements
Future State *Paint a picture* *of the ideal way of working*	• Marketing can take ownership of experimentation program (Brand Marketing, Demand Gen, Digital Marketing) • High-velocity (faster iteration) experimentation • Personalization at scale
Positive Outcomes *High-level benefits*	• Create, test, and roll out personalized feature delivery, multiple experiments, and multi-armed bandit campaigns modifying feature variables pre-defined by engineers, without requiring a developer • Make changes to your website without requiring a developer with easy-to-use WYSIWYG visual editor • Add new behavior to pages using extensions • Hotfix bugs without developer involvement • Test and iterate before you build in code • Test and iterate on newly released features by your non-developer teams

Metrics
 	
Business objectives tied to this initiative
	• B2B
	   o No. of leads
	   o Quality of leads
	   o Lead conversion rate
	   o Newsletter Subscriptions
	   o …
	• eCommerce
	   o Revenue
	   o Average Order Value
	   o Conversion Rate
	   o …
	• Media
	   o Pages / Session
	   o Ad Impressions
	   o Subscription conversion rate
	   o …
	• B2C Software (Streaming, …)
	   o Subscription Conversion Rate
	   o Churns Prevented
	
	• Velocity – number of test launched
	• Conversion (online & offline) – revenue, sign-ups, downloads
	• UX Metrics – Time spent on page, click rates, bounce rates, scroll rates
	• Time to launch an experiment
	• Team efficiency: Saves Dev time

	• Time to value: needs low training to start launching tests
Required Capabilities *Core products, enhancements & features that enable the "After"*	• Visual Editor • JS Code Editor • Stats Engine • Scaled Setup (Pages, Extensions, Events) • Deep targeting capabilities (cookies, JS, IP, Geolocation, behavioral targeting, ..) • Extensions • Visual Tracking • Performance Edge

Differentiators
 	
What uniquely positions us against the competition "How we do it Better"
	• Performance Edge
	   o Fastest A/B Testing solution on the market in terms of performance impact on websites
	• Stats Accelerator
	   o Unique on the market. Enables getting to results of experiments much faster, enabling faster decisions
	• Stats Engine
	   o Best sequential stats model that balances speed to a result with preventing negative business decisions
	• Scaled Setup (Pages, Extensions, Events)
	   o Enable rapid experimentation through templated experiments and setups
	• Visual Tracking
	   o Track events by selecting elements and pages to be tracked. No help from development needed
	• Fully featured REST API
	   o Automate everything that can be done through the UI for additional automation and scale
	• Github Integration
	   o Developers can directly integrate with Github
	• We win because we can address the objections to adopting browser-based testing tools: Latency (Perf. Edge), Complexity to implement (extensions, Rapid Exp. Team)
	
	Competitor notes:
	
	Front-end capabilities are good differentiators against back-end-focused competitors like LaunchDarkly, but competitors like VWO, Convert, AB Tasty, and Dynamic Yield are on par with the core capabilities. Differentiation comes from identifying what the customer needs and positioning our differentiated features.

Target Audience *Relevant buyer & user personas by title/seniority*	**CMO/Marketing Manager** • Decision making authority over marketing tech stack • Someone who owns the digital experience, or own the goal of improving KPIs related to the digital experience • Typically, multiple teams (Brand Marketing, Demand Gen) who are all results focused and want stronger tools that help optimize output • Can be marketing, UI/UX, Digital, or Ecommerce teams

	• (If talking to a product manager that owns digital experience, please see Feature Experimentation) **Characteristics** • Digital maturity – Mid and up • Business priorities/Focus areas – improve engagement, conversions, or satisfaction. Increase customer insights • Other tech ownership may include tag managers, analytics & heatmapping tools, voice of customer • Possibly new to experimentation / first-time buyer • Might have used at other orgs before • Might be familiar with experimentation adoption from product team, but want to now "enable" marketing team **Other Titles:** • eCommerce Manager • Optimization Manager
Proof Points *Customer Stories,* *AR & Data Points (e.g. # of experiments run/year)*	[Crate and Barrel](#) - Went from running 30 tests/year with 0 personalization to 100 tests/year amongst which include personalization - +6% CVR - +20% RPV - -10% Bounce Rate "It all leads to more tests and more experimentation heroes. And we are constantly sharing not just results, but the entirety of the process. That can fundamentally affect the short- and long-term success of any program." - Christine Garvey, Senior Manager of Personalization and Optimization at Crate and Barrel "Scaled personalization could unlock $1.7 to $3 Trillion in value", McKinsey Global Institute, Q2'18 "Increased developer productivity delivered $1.5 million in savings", Forrester, Q4'21
Directional Discovery *Leading questions that tie pain to capability*	• "How are you measuring the impact of changes on your website today?" • "Who makes decisions about what goes live on the website?" • "Do you have access to developers that help setting up A/B tests (a no is usually a red flag, since Web also requires some help from devs)" • "Is your organization doing A/B testing outside of your core product team?" • "Have you personally used or adopted experimentation in the past?" • "What are some basic use cases that your team is interested in testing on?" • "Can you share other items in your marketing tech stack that might include relevant data or segmentation that would help power your experimentation? Including – Marketing Automation, 3rd-Party Data analytics, CRM" • "If your product team or developers currently manage experimentation, can you share your current process for creating new programs?"

FEATURE FLAGGING / FEATURE MANAGEMENT

Solution	**Experiment**
Use Case	**Feature Flagging / Feature Management**

Products	Feature Experimentation
Summary	Feature Experimentation enables development teams to decouple releases from deployments through Feature Flags and Rollouts, reducing risk for failed deployments. FX enables teams to adopt trunk-based development, pushing code live whenever it is ready without losing control of the release process. Through feature configurations and FX's Rules Engine, arbitrary Feature Variations can be rolled out by product management or engineers through experiments or targeted deliveries, controlling releases and enabling data-driven decision-making. FX combines feature management and experimentation into a single workflow enabling development and product management to work closer together and build more impactful features.
Before State *Key challenges and problem statements*	**Product Management** - A/B tests are separated from code deployment because not run on top of feature flags **Engineering / Development** - Bugs have a large impact due to long time to fix / difficulty to roll them back. Downtimes due to bug fixes - Feature deploy and release must be synchronized - Difficulty managing code branches since features have to be deployed at release time - Engineers needing to be online to do so at inconvenient hours - Difficult to progress features through development environments
Neg. Consequences *The result of the "before"*	**Product Management** - No flexibility to respond to user feedback or issues / takes a long time to respond - Revenue loss due to bugs and customer dissatisfaction - Increased alignment/communication effort between Engineering and PM **Engineering** - Lower developer productivity - Loss of control over release process - Increased alignment need with product and release processes - Doesn't allow SE teams to have a margin of error in case the feature/change is unsuccessful\ - Larger blast radius for bugs
Future State *Paint a picture* *of the ideal way of working*	**Product Management** - PMs quickly make changes to key feature variables to respond to user feedback or issues - PMs release features to select audiences gradually without involvement from engineering **Engineering** - Problematic features are quickly disabled / rolled back without impact on the overall application - Code is deployed when it's done without having to release it immediately. Enables trunk-based development **Both:**

	• Features are gradually rolled out to users, allowing teams to monitor and assess their impact and make adjustments before fully releasing them and reducing risk
Positive Outcomes *High-level benefits*	**Product Management** • PMs and Engineers can release to a subset of users (pre-defined segments) progressively at any point in time **Engineering** • Engineers can merge and deploy code at any time without having to actually release features or changes, resulting in faster releases with less risk • Engineers can deploy at any time without adhering to common release schedules, which reduces the need for alignment • Build kill switches to turn off features that are not performing well
Metrics *Business objectives tied to this initiative*	• Reduced number of failed deployments • Lower impact of bugs on revenue • Faster development cycle times • Increased developer productivity • Lower cost / feature • Increased experimentation velocity
Required Capabilities *Core products, enhancements & features that enable the "After"*	• Feature Flags • Targeted Deliveries • Progressive Rollouts • Scheduled Rollouts (In Dev: Q1 release) • SDKs • Agent (Microservice) • Environments • REST API
Differentiators *What uniquely positions us against the competition "How we do it Better"*	• Rules Engine ○ Multiple deliveries / flag • Stats Engine (our sequential stats model is great for engineering metrics like error rates etc.) • REST API • Native CDP Integrations **Competitor notes:** - LaunchDarkly and Split.io are ahead in feature management, but if you also need experimentation, Optimizely is a better combined fit. We have the core feature set you need for feature flagging and far better experimentation capabilities - It will be difficult for us to compete on Feature Flagging only use-cases but should seek combined deals or position our free feature flagging product to upsell the customer later

Target Audience *Relevant buyer & user personas by title/seniority*	**Buyer Personas** **CTO** - Oversees product teams and development across org - Manages development pipeline including deployment **Characteristics** - Digital maturity - high - High focus on ROI, cost of tech stack - Engineers - Operations - Secondary Architects - VP of Engineering - Director of Engineering **User Personas** - Software Engineer - App Developer - Web Developer
Proof Points *Customer Stories,* *AR & Data Points (e.g. # of experiments run/year)*	Blue Apron - 10x increase in experiments run per month after adopting Full Stack - 100% number of product teams using experimentation - 37% decrease in customer issues related to unexpected orders "With Optimizely Full Stack, Blue Apron gets statistically valid experimental results on their server and mobile applications which helps them make faster, more informed business decisions. With Optimizely's real-time results, Blue Apron was able to avoid launching a redesigned reactivation flow, which would have resulted in a significant revenue loss. Blue Apron can now quickly and easily build and set up new tests that were simply not possible before With a consistent framework in place, the number of people at Blue Apron now performing testing has expanded well past the Growth/Member Experience team. Anyone in the company on any team can create a test, see the results, analyze it and use that data to inform their decisions." Ambassador Theatre Group (unreleased case study – **INTERNAL USE ONLY**) - 2 experiments run weekly - More than 10% increase in overall checkout conversion "To ensure controlled experiments, the Product Insight Manager of ATG, Monty Mohamed, shared that Optimizely's Targeted Rollouts function has helped the team immensely, especially Feature Flags.

	Feature Flags enable ATG to canary test with quicker cycles in a real-world setting without making disruptive code changes. ATG can also quickly roll back tests by a toggle of a button when it is set up with a Feature Flag, easing the work of developers and engineers."
Directional Discovery *Leading questions that tie pain to capability*	"Do you current adopt feature flag or rollout capabilities?" "If you do, what do you use? How is your adoption?" "If you don't, have you considered it in the past?" "Have you considered the impact of adopting experimentation into your development cycle and release planning?"

WORKFLOW MANAGEMENT / COLLABORATION

Solution	**Experiment**
Use Case	**Workflow Management / Collaboration**
Products	**Web Experimentation / Feature Experimentation + Welcome**
Summary	Best practices start with efficient collaborative tools, shared vision and goals. Manage your experiments through advanced seamless program management specifically built for Web and Feature Experimentation. Get an integrated hub for capturing ideas, prioritizing hypotheses, and managing experiments easily across all stakeholders in your company. Scale experimentation and increase program velocity through improved collaboration and knowledge sharing.

Before State
 	
Key challenges and problem statements
	Program Management
	• **Themes:**
	    ○ Governance
	    ○ Project Management
	    ○ Ideation and Experiment Strategy
	    ○ Transparency and collaboration
	    ○ Knowledge and Results Sharing
	    ○ Executive visibility and org-wide visibility
	• As organizations grow larger: it becomes more complex to unify best practices and collaborate to build experiences more effectively
	• Organizations need to be armed with the right tools to build a culture of experimentation
	• Increased velocity entails unnecessary increased project management - until the program manager becomes the bottleneck
	• Lack of governance: teams of different maturities operating independently resulting in inconsistent outcomes and difficult to trust insights
	• Lack of planning and no shared expectation of timelines; work is completed at the earliest convenience meaning rarely at all.
	• Organizations building a program for the first time, depend on outside expertise and a high learning curve to get started

	• Pressure to increase velocity with no resource planning results in straining certain resources while creating slack for other teams to create poorly optimized team.
	
	Product Management
	• Coordinating running an A/B test across multiple teams is challenging
	• Scaling experimentation across multiple product teams is challenging
	• If coming from existing solution:
	    ○ Difficult process / high cost to set up experiments
	    ○ Low experimentation velocity (number of experiments run) leading to limited value from existing program

Neg. Consequences *The result of the "before"*	**Program Management** • Inefficient allocation of time and resources • Difficult real-time collaboration • No way to share and reference key learnings and treat them like a long-term company asset • Inefficient and unscalable processes **Product Management** • Low experimentation velocity (number of experiments) • Development resources wasted building capabilities that don't move the needle
Future State *Paint a picture* *of the ideal way of working*	**Program Management** • Best practices start with efficient collaborative tools, shared vision and goals • Get an integrated hub for capturing ideas, prioritizing projects, and managing experiments • Different groups across your company can collaborate with ease • Increase program velocity by up to 5 times through improved collaboration and knowledge sharing **Product Management** • Scale experimentation across your organization
Positive Outcomes *High-level benefits*	**Program Management** • Use test plans to capture all the information you need to develop an idea into an experiment **Product Management** • Use test plans to capture all the information you need to develop an idea into an experiment • Test roadmaps to get a holistic view of running and planned experiments on either a list view, timeline view or board view **Engineering** •

Metrics *Business objectives tied to this initiative*	• Experimentation Velocity (No. of experiments run) • Experiment quality • Win Rate • Learning Rate (% of experiments with a significant result) • Impact – Stat sig positive negative rates • Culture – number of teams testing • Contribution – number of employees submitting test ideas
Required Capabilities *Core products, enhancements & features that enable the "After"*	**Feature Experimentation or Web Experimentation** - "Experimentation Collaboration" (TBR: H1'23)
Differentiators *What uniquely positions us against the competition "How we do it Better"*	• Optimizely Content Marketing • Web • Feature Experimentation • Optimizely results page with Stats engine • Onboarding & Expert Services • Rapid Experimentation team
Target Audience *Relevant buyer & user personas by title/seniority*	**CTO** • Oversees data decisions in the org • Wants to invest in tools that can access single-source of data, regardless of tool • Oversees Data Team & Tech Stack Characteristics • Digital maturity – high • Business priorities/Focus areas – improve integration of tools, improve data quality **CMO** • Drives conversion and optimization across marketing teams • Wants to make personas and personalization a high priority Characteristics • Digital maturity – high
Proof Points *Customer Stories,* *AR & Data Points (e.g. # of experiments run/year)*	HP: - Focused on collaborative experimentation - Implemented weekly program reporting to share experiment findings - Ran ≈ 500 campaigns - Drove an incremental $21 million in revenue
Directional Discovery	**Questions leading to promoting Program management in Experimentation:** • What does your collaboration process look like between departments?

	How many different teams work on your experimentation practice?
Leading questions that tie pain to capability	• How many different teams work on your experimentation practice? • If you could collaborate more teams into your proposed experimentation practice and governance, who would those be?

PERSONALIZATION

Solution	**Experiment**
Use Case	**Web Personalization**
Products	**Web Experimentation**
Summary	With all the same benefits as our Web A/B Testing use case, go beyond pure experimentation and create curated short- or long-term experiences for your users. Take advantage of our powerful WYSIWYG tools like the Visual Editor and our easy-to-use OOTB audiences and integrations to provide your users hand-crafted designs at a fraction of the normal cost and effort. Don't stop at understanding: activate your user data to drive engagement and website KPIs by Measure the impact of personalization efforts, target and optimize experiences to audiences built from your own data and use website behavior to include visitors in your experience.
Before State *Key challenges and problem statements*	**Web Development** • Web development tries to hook up several vendors and analytics systems to enable marketers to target campaigns to users to drive conversions • Difficult to personalize website using different segmentation and activation vendors • Unable to rollback unsuccessful campaigns easily, usually requires code deployments **Marketing** • No ability to execute personalization campaigns and strategies without extensive development • Unable to measure outcomes from personalization activities, makes ROI unclear **Analytics** • Analytics data is too far removed from the personalization, requiring custom querying and adds a lag to insights • Requires high level of data analysis skills to fully understand
Neg. Consequences *The result of the "before"*	**Web Development** • Extra work required to constantly create personalized experiences, including building proper targeting (which can be very difficult) • On the hook for rolling back breaking changes, requiring more availability from production team **Marketing** • Bottlenecked by engineering, who have competing priorities • Slow turnaround time on authoring and publishing new experiences **Analytics** • KPI movements are difficult to tie to personalized experiences and even more difficult to trust (low confidence in results being accurate)

Future State *Paint a picture* *of the ideal way of working*	**Web Development** • No longer necessary to push code to rollback campaigns – not even necessary to be involved in the process (non-technical users can now do so without help) **Marketing** • Able to use WYSIWYG tooling to author and publish personalization campaigns with strong audience targeting with little to no developer involvement • Higher velocity of KPI optimizations **Analytics** • Strong confidence in the outcomes of a personalization campaign • Easy to understand results
Positive Outcomes *High-level benefits*	**Web Development** • Able to focus on higher priorities **Marketing** • Easily create and deliver personalization campaigns to drive improved website KPIs **Analytics** • Use targeted personas as part of experiments to evoke validation or analysis at specific levels

Metrics
 	
Business objectives tied to this initiative
	• B2B
	   ○ No. of leads
	   ○ Quality of leads
	   ○ Lead conversion rate
	   ○ Newsletter Subscriptions
	   ○ …
	• eCommerce
	   ○ Revenue
	   ○ Average Order Value
	   ○ Conversion Rate
	   ○ …
	• Media
	   ○ Pages / Session
	   ○ Ad Impressions
	   ○ Subscription conversion rate
	   ○ …
	• B2C Software (Streaming, …)
	   ○ Subscription Conversion Rate
	
	Churns Prevented

Required Capabilities *Core products, enhancements & features that enable the "After"*	• Behavioral Targeting • Cookie-Based Targeting • Real-time segmentation • Audience Upload
Differentiators *What uniquely positions us against the*	• Connectors with third-party data aggregators (Tealium, Adobe Audience) • Dynamic Profiles to bring in your own data • Integration with native CDP for advanced targeting • Results page integrates with audiences to break down results by persona

competition "How we do it Better"	• Personalization Campaigns to target against multiple personas in same test
Target Audience *Relevant buyer & user personas by title/seniority*	**CMO** • Focused on demand or brand marketing, has requirement to optimize marketing campaigns • Data focused – understands personas, ICP and wants to assure tests can validate different groups **Characteristics** • Digital maturity – high • Business priorities/Focus areas – improve marketing ROI, improve data quality
Proof Points *Customer Stories,* *AR & Data Points (e.g. # of experiments run/year)*	Missguided: • Experiment: Premium service offering, personalized offers for VIP customers • Delivering a personalized shopping experience to every visitor is contributing to Missguided's growing customer base • +177% conversion uplift • +33% relative increase in revenue through personalization
Directional Discovery *Leading questions that tie pain to capability*	**Questions leading to promoting Personalization in Experimentation:** What does personalization mean to your organization? How defined are your organizational personas? Where is your "golden record" for your customer? Do you have a source of truth for segmentation?

Solution	**Experiment**
Use Case	**Feature Personalization**
Products	**Feature Experimentation**
Summary	FX customers will get 25+ pre-built audiences out of the box because ODP comes with pre-built real-time segments (RTS). FX customers with the FX x ODP integration will be able to segment anonymous users based on their past behavior and attributes collected while previously using the same device or if they've ever logged in. They will also be able to target large lists of users by an identifier for features or experiments in FX. Using ODP data, customers will be able to roll out personalized experiences on top of feature flags using targeted rollouts and MAB in combination with ODP.
Before State *Key challenges and problem statements*	**Product Management** • Want to build complex/specific audiences from more first- and third-party data sources, target lists of users their IDs, segment anonymous visitors, and have a portable source of truth for their audiences between FX projects, FX+Web, and for use in third-party software

	• Want to reduce the developer effort involved in sending user attributes for audiences **Engineering / Development** • Developer resources required to enable targeting by user attribute(s) • Prefer not to gather and send user attributes at run-time to our SDKs for audience targeting purposes and would instead prefer that come from a CDP (like ODP) **Marketing** • Want to add channel(s) to their multi- and omni-channel marketing campaigns with a single source of truth for audiences **Data Science** • Wants to apply data-driven segmentation to features to analyze who is using what features and how adoption rates attach to ICP
Neg. Consequences *The result of the "before"*	**Product Management** • Customers may invest significant effort to build or buy 3rd-party analytics & segmentation tools **Engineering** • Must build in-house tools for persona-based feature activation • Cannot manage distribution of features to different cohorts and instead must resolve to more broad feature rollouts **Marketing** • Customers might have to have different segments in different tools, causing disconnect between "view" of customer
Future State *Paint a picture of the ideal way of working*	**Product Management** • Fast and accurate audience evaluations within Web and Feature Experimentation **Engineering** • Reduction in internal costs by using external tools for managing features • Can enable organization without internal backlog to build customer personas that use gated features **Marketing** • Customers might have to have different segments in different tools, causing disconnect between "view" of customer
Positive Outcomes *High-level benefits*	**Product Management** • Segments can easily be defined from multiple first and third-party data sources, enabling a single source of truth for audiences across first-party, Optimizely, and 3rd party products • Centralize customers' audiences in one place and target them with Feature Experimentation, Web Experimentation, and other 3rd party tools • The definition of a customer segment is unified into a CDP: instead of having various systems with different ways of understanding what a "recent customer" is, all that data is captured AND defined in a CDP - more datapoints into the CDP (attributes, behaviors, predictions) allow more fine-grain segmentation **Engineering** • Faster experimentation velocity and more accurate audience evaluations

	• Lower developer effort to provide underlying data used to segment **Data Science** • Better product analysis for customer organizations to review adoption of product across different personas • Better telemetry and quality of data to give back to the organization on product usage, margin etc

Metrics
 	
Business objectives tied to this initiative
	B2B
	    o No. of leads
	    o Quality of leads
	    o Lead conversion rate
	    o Newsletter Subscriptions
	    o …
	• eCommerce
	    o Revenue
	    o Average Order Value
	    o Conversion Rate
	    o …
	• Media
	    o Pages / Session
	    o Ad Impressions
	    o Subscription conversion rate
	    o …
	• B2C Software (Streaming, …)
	    o Subscription Conversion Rate
	
	Churns Prevented

Required Capabilities *Core products, enhancements & features that enable the "After"*	**Web Experimentation and/or Feature Experimentation** • ODP • Real-time Segments (RTS) • Experimentation – Specific features: Custom attributes and technology integrations • Page builder, audience builder, Visual Editor, one click event tracking, metrics builder • Extensions • Optimizely results page with Stats engine • Onboarding & Expert Services Rapid Experimentation team
Differentiators *What uniquely positions us against the competition "How we do it Better"*	Native CDP integration Customer Attributes and DCP Mutual Exclusion capabilities Ability to target multiple audiences per campaign and prioritize for users who qualify for more than one audience with Optimizely Personalization **Competitor notes:** Best used against server-side competitors (e.g. LaunchDarkly) who are behind on targeting capabilities

Target Audience *Relevant buyer & user personas by title/seniority*	**CTO** - Oversees data decisions in the org - Wants to invest in tools that can access single-source of data, regardless of tool - Oversees Data Team & Tech Stack Characteristics • Digital maturity – high • Business priorities/Focus areas – improve integration of tools, improve data quality **CMO** - Drives conversion and optimization across marketing teams - Wants to make personas and personalization a high priority Characteristics • Digital maturity – high Other target roles: • Product Managers • Marketers • Analytics Characteristics: Not new to experimentation, and decent handle on customer data, access to analytics/data science/database management resource
Proof Points *Customer Stories,* *AR & Data Points (e.g. # of experiments run/year)*	**Note**: Integration between ODP and Web Ex/Feat Ex is still maturing. Only one customer currently scheduled for Beta in Q42022 Increase in conversations on CDP, Personalization and segmentation from new prospects – customers want to talk about this and by solving it "better" we resonate with their needs
Directional Discovery *Leading questions that tie pain to capability*	• "Do you have focus on Personalization in your org?" • "Where do you store your customer profile data?" • "Have you invested in a CDP in the past?"