

Introduction

Product development is difficult. That's why we developed [The Product Thinking Playbook](#), a mapping tool that helps multi-disciplinary teams develop an execution strategy for ambitious new products. Consisting of various tactics and techniques borrowed from Design Thinking, Agile Development, Lean Product Strategy, and Jobs-To-Be-Done Theory, the Playbook helps facilitate conversations about what you will and (just as importantly) will not do to achieve your product goals.

Upon completion of a Playbook exercises, the final “map” becomes your strategic execution plan: a co-created consensus that ensures alignment on purpose and approach. The Playbook's flexibility allows stakeholders to periodically reevaluate approaches, find inspiration, and “get unstuck”.

The Product Thinking Playbook is neither prescriptive nor comprehensive, but exists to help teams find the right path forward—one crafted to your unique context, capabilities, and aspirations. It is not about best practice, but better outcomes and better products.

Instructions

A Playbook exercise has two goals. The first is to create alignment through dialogue about what will and will not be done to achieve the desired outcome. The second is to generate a physical artifact that represents the execution strategy that can be referenced—and modified—throughout the life of the project.

Attendees

Key stakeholders
Execution team

Duration

60 Minutes

Part 1 : Share Context

30 minutes

Tell the story of why this initiative is important. What are the goals of this project? What is it trying to accomplish? If you are validating what to build, describe the problem space. If you have a validated concept, explain why the product should exist as well as what impact you hope it will have on your business and the end user. Leverage any existing research.



Part 2: Playbook Mapping

30 minutes

Map your execution plan using the Playbook cards in the following order:

1. Tactics

Tactic cards represent a set of methods and strategies that can be used to accomplish desired outcomes.

Lay them out sequentially. If activities overlap, you can stack them to create parallel tracks.



2. Discovery and Delivery Tracks

Discovery and Delivery cards label parallel tracks according to their main focus. Discovery tracks are geared toward learning and validation while Delivery tactics are geared toward design and development.



3. Iteration Loops

Identify the start and end points for a repeated set of tactics, where applicable.



4. Milestones

Add milestone cards at the conclusion of critical tactics to update or demonstrate progress.



5. Techniques

Technique cards specify how you will execute against your chosen tactics in greater detail.



Can't find the right card?

Use this card as a substitute for any missing
Tactic, Technique, or Milestone.



Can't find the right card?

Use this card as a substitute for any missing
Tactic, Technique, or Milestone.



Discovery

Delivery









Accessibility Audit

Review the accessibility of a product or design. Methods include manual and automated accessibility testing as well as usability testing with individuals possessing a variety of impairments.

Accessibility Implementation

Evaluate and implement design practices and technologies to ensure the product is accessible and usable by diverse user groups including, but not exclusively, impaired individuals. Define accessibility specifications and supported assistive technologies for end-to-end product testing.

Technique

Affinity Mapping

Organize research observations, evidence, and/or ideas into thematic groups to uncover patterns and deeper insights.

Agile Rituals

Throughout the entirety of a project, and especially for the Delivery track, teams use rituals to help reinforce the values of Agile. Rituals include backlog prioritization, team retrospectives, iteration planning meetings, story estimations, and daily standups.

External Technology Exploration

Determine what interfaces, such as APIs and SDKs, and development environments are available for a particular technology. Inspect the feature set and test capabilities; document and communicate findings.

Back-End Implementation

Implement services for business logic and data handling to support devices or a front-end implementation.

Business Model Development

Document an existing business model or explore new business model possibilities. Using the Business Model Canvas as a template, model how a business delivers value to customers.

Card Sorting

During research or testing, users rank, order or cluster a set of features, jobs, pains, gains, or procedural steps to represent their priorities, values, and overall thinking process in relation to an existing product or potential concept. Card Sorting informs information architectures, user journeys, and value proposition design.

Cohort Analysis

Compare the behavior of different customer cohorts over time to identify high-engagement and low-engagement cohorts and learn what factors are driving this variation. Cohort Analysis helps isolate performance metrics that can be masked by overall growth.

Competitive Review & Trend Analysis

Examine an industry to identify relevant trends in technology, business, and design. Assess the competition to identify feature sets and potential differentiators by exploring adjacent categories.

Gatekeeper Collaboration

(Compliance and Risk Collaboration)

Engage and collaborate with Legal and Security teams to define regulatory and compliance requirements early in the product development process. Establish ongoing reviews and audits during development to mitigate new risk areas.

Concept Evaluation

Evaluate initial concepts against an established set of criteria, such as value to users, to the business, or technical investment required. Each criteria may be assigned a score or weighted to enable concept prioritization. Prioritized concepts will be the focus for further refinement, prototyping, and testing.

Conversation Flow Map

Chart the flow of a user's interaction with a conversational interface (chat or voice). Identify utterances, intents, responses, and conversational repair.

Customer Profile Development

Clarify understanding of user segments by mapping their jobs, pains, and gains. Ensure jobs, pains, and gains align to a particular product experience or opportunity area, as informed by research. Data points should also be validated through testing.

Data Audit

Collect existing internal and external data sets and perform initial explorations to assess their size, relevance, availability, and suitability.

Data Engineering

Implement the data architecture to collect, transform, store, and retrieve data, typically on a distributed platform, for service implementation, advanced analytics, archiving, and/or modeling purposes.

Design Sprints

For each feature, articulate the problem or opportunity, sketch possible solutions, define the workflows, and create prototypes or production-ready designs. Sprints may also include user testing. Designers focus on one feature during each weekly sprint.

Design System

A set of reusable assets, rules, and specifications for a user interface. It demonstrates how to build or extend a product to ensure a cohesive and uniform visual language.

DevOps Implementation

Implement practices and technologies to automate and streamline the development and operation of software products and infrastructure by enabling continuous integration, continuous delivery, and deployment.

Diary Studies

Diary Studies is a research method that collects qualitative information by having participants record entries about their everyday lives using a journal or an online service in either written, survey, or video form. Entries can be about the activity, product, or overall experience.

Dogfooding

Conduct internal testing and collect quality and actionable feedback from internal product development teams or other targeted units within the organization. Dogfooding generates empathy with the end user's pains and heightens the development team's sense of product ownership.

Exploratory Testing

Conduct unstructured QA testing to identify quality/UX failures outside of planned scenarios and discover edge cases not easily covered by the defined QA workflow.

Feature Flags

Release features to a subset of users/visitors to test their performance. This method is typically used on existing products or MVPs.

Technique

Front-End Implementation

Implement the user-facing interface and experience of an application based on UX/UI designs.

Funnel Analysis

Examine data that measures how users are navigating through a product to reach a desired conversion point. Funnel Analysis uncovers where users drop off and prioritizes where to dedicate efforts to improve conversion rates.

Gated Rollouts

Release product versions through phased and/or parallel deployments to a subset of users and/or infrastructure so production risks can be mitigated by safe rollbacks before a full-scale launch. Every deploy phase can be gated based on incremental feedback.

Horizon Scanning

Detect early signals of change in core and peripheral environments to develop strategies for the mid- to long-term future, using STEEPV (social, technological, economic, environmental, political, values) as a framework to uncover actors and drivers of change.

Hypothesis Driven-Validation

Identify and list all the known assumptions about the product concept and its intended user. Rank them in order of importance (based on how critical they are to the success of the product and/or their relative risk) and identify appropriate methods to validate or invalidate these assumptions one by one.

Ideation

Conduct a series of activities to generate, evaluate, and prioritize possible product directions, concepts, and/or components. The resulting ideas and the common patterns between them become inspiration for further concept development and validation.

Incident Response Management

Design and implement processes to orchestrate the appropriate response, escalation, and resolution to production incidents. This includes active monitoring, established communication channels, guided assessment of severity levels, clear escalation paths, and effective assignment of designated responders.

Infrastructure Design

Generate options for the product's technical infrastructure and validate which approach will be the most effective solution for anticipated initial needs and growth factor. Activities may include prototyping and evaluating alternatives for data ingestion, storage, and processing, which can be used to enable impactful AI, ML and Data Science outcomes.

Internationalization and Localization

Evaluate and implement design practices and technologies to enable flexible localization of the product for audiences in target regions / markets. Define multi-regional specifications for end-to-end product testing.

Minimum Viable Product Definition

Define the minimal version of a new product that demonstrates the core product value. Using key insights and opportunities, product vision, and goals as a starting point, the product team identifies, prioritizes and translates core features and solutions into a backlog to achieve product outcomes. The MVP release should allow product teams to iterate rapidly through user learnings and validated product assumptions.

Multivariate Testing

Test two or more variables with users to see which is preferred and/or performs more optimally in achieving its purpose.

Operations Runbook

Document execution steps to follow when responding to production incidents, and running routine operations and maintenance tasks. Runbooks translate cross-team knowledge and continuous learning into actionable steps so teams can autonomously respond to incidents and maintain product operations. Repetitive Runbook steps are often automated or templated to reduce human error.

Performance and Error Monitoring

Implement technologies and processes to streamline real-time monitoring, tracing, and alerting of errors and performance bottlenecks across the tech stack. This might include capabilities such as enriched stack traces, distributed querying of event data, visualizations of infrastructure resources, etc.

Product Analytics Review

Review product data to understand user behavior and measure performance against key business objectives. Analytics Reviews help uncover key insights, identify missing data points, and determine actions for improvement.

Product Backlog

Create a prioritized list of user stories, tasks, and/or bugs necessary to complete a release to ensure that the product team focuses on the most relevant tasks in an optimal sequence.

Product North Star

A living document that connects the strategic purpose of the project to the team's tactical activities. This document is informed by ongoing research into user jobs, pains, and gains as well as the strategic importance of the product to key stakeholders.

QA Requirement Analysis

Beginning with a validated solution backlog, QA analyzes product requirements, design specifications, and system interactions to validate accuracy, comprehensiveness, and testability. With this understanding, test scenarios and cases are created and prioritized on a sprint-by-sprint basis.

QA Testing Strategy

Define the test-execution approach and plan that outlines the test scope, objectives, and processes to ensure product quality and QA effectiveness. The strategy should determine the appropriate framework, test levels and phases, device/platform and code coverage standards, and environment management process to ensure early issue detection and resolution. Depending on the depth of coverage, the scope may include performance, scalability, and security testing.

Scenario Building

Create scenarios for potential futures by researching signals of change in business, culture, and technology. Illustrate these scenarios through storytelling and use them to stress-test current initiatives, iterate on potential strategies for the future, and identify implications for the business.

Security Review

Document information contained or used by the product and observe its flow through the product and connected systems/devices. Compare the security, authentication, and authorization applied at each interface or storage point to industry best practices.

Service Blueprint

Document how the product fits into a broader ecosystem and map business operations across the entire user journey. Pay particular attention to “front stage” touchpoints that the user sees, and the “back stage” and “behind the scenes” processes that are largely invisible.

Software Architecture

Establish multiple options for the software architecture—following commonly defined patterns—and validate which approach will be the most effective and extensible solution.

Stakeholder Interviews

Interview key stakeholders to better understand the objectives, goals, current/future state of the industry, and the business through multiple perspectives. Synthesize these inputs to further refine the definition of product and project success, and to present a more holistic and nuanced strategy for product development.

Storyboarding

Illustrate a story or sequence of events to explore the beginning, middle, and end of how a user interacts with a product, service, or AR/VR experience. In some cases, storyboards may be translated into concept videos to increase narrative fidelity.

Surveying

Issue a questionnaire to a targeted group of people to identify common problems, behavioral patterns, and perceptions. Surveys can also be used to validate proposed product directions.

Technique

Technical Audit

A deep dive into the current tech stack, looking closely at what tools and services are used and how hardware and software are integrated.

Technical Feasibility Review

Determine the technical feasibility of a concept or prototype given project constraints, available technologies, and business requirements.

Technical Performance Review

Evaluate the technical performance metrics of a product across targeted devices and use cases. Sample metrics include responsiveness, memory usage, battery usage, load time, and peak throughput.

Technical Review and Collaboration

Conduct peer or group technical reviews to collaborate on architecture design and evaluate code quality. Collaboration practices include pair programming, cross-platform code reviews, and architecture proposals. These practices promote continuous code review, deeper reflection of implementation approaches, knowledge transfer across technical domains, and shared code ownership across the team.

Test Automation

Automate test execution, test data management, and issue reporting to ensure product and software quality at all times. Depending on the level of automation required and trade-offs with overall testing efficiency, the automated test suite can encompass code analysis, unit tests, cross-platform functional tests, regression tests, etc.

Usability Testing

Evaluate the efficiency, effectiveness, and enjoyability of an interface by testing it with users, ideally in the context most applicable to its purpose. Usability Testing helps identify issues that may prevent users from successfully interacting with the interface.

User Feedback Review

Collect and review explicit customer feedback to add qualitative data to the team's product performance analysis. Sources include support tickets, call centre data, and app store reviews.

User Interviews

Conduct guided conversations with existing and/or target users, as well as other relevant individuals on the demand side to learn about their experiences, understanding, struggles, and feedback directly related to the area of investigation.

Experience Mapping

Map the experience of a product, prototype, or process to help understand and address obstacles, needs, and opportunities.

Detail the actions relevant actors (e.g. users, business stakeholders) take and the changes to how they feel at key touchpoints in the experience.

User Story Mapping

Map a product's user goals in a sequential order to illustrate the tasks and activities that they might perform to help them achieve their goals. This helps define and prioritize product features for MVP release and beyond.

UX Audit

Review an existing product to identify problems in the user interface or overall experience according to recognized usability principles and current best practices.

Value Proposition Development

Map the “pain relievers” and “gain creators” that your product delivers to the identified pains and gains of a Customer Profile Canvas. The Value Proposition Canvas summarizes the fit between the product and the job the customer is trying to make progress on.

Wireframing

Sketch a low-fidelity version of the product's visual interface to guide and establish early-stage design decisions.

Wizard of Oz Testing

Complement an early-stage prototype with human elements during testing to more realistically simulate the proposed solution, revealing more representative behaviors from the users testing it.

Architecture Exploration

Perform research and ideation activities to identify potential designs for future-state application and system architectures, and options for software solutions and frameworks that can be used to achieve the intended technical outcomes.

Manual Testing

Conduct manual software testing based on defined test cases and scenarios in the perspective of an end user. This is suitable during Exploratory, Usability, and/or ad-hoc testing for non-repetitive tests that require human observation.

Information Architecture

Describe the different content pieces and how they relate in a system to get a handle on the volume and complexity of the content. Activities include content inventory, information grouping, taxonomy development.

Tactic

Agile Development

Develop software using a methodology designed to improve quality and responsiveness through short development cycles and frequent releases.

Tactic

Analytics Implementation

Evaluate, select, and implement product metrics tools to capture and track data on product activity and user behavior.

Tactic

Business Research

Conduct research to develop a thorough understanding of the business objectives and strategy by employing techniques such as competitive review, horizon scanning, or stakeholder interviews. The outcomes uncover relevant business viability risks considered during opportunity assessment.

Tactic

Concept Generation

Generate, define, and evaluate multiple product concepts that leverage key insights and areas of opportunity uncovered through research.

Tactic

Delivery Program Planning

Organize and manage the Agile software development process based on project goals. This includes product and project operations such as scope management, product performance analysis, process improvement, team organization, project tooling, and stakeholder alignment throughout the implementation to ensure continuous learning and delivery efficiency.

Tactic

Demand Validation

Test the desirability of a proposed strategy, concept, or feature against the problem or opportunity it is intended to address. The goal is to ensure that the solution addresses a real, observed struggle and/or generates new demand.

Tactic

Go-to-Market Planning

Define the technologies, activities, and orchestration required to carry out the go-to-market strategy of the product launch. This is commonly revisited during product development to adapt to strategic changes as new market risks and opportunities emerge.

Tactic

Prototyping

Iteratively design and/or develop representations of concepts and experiences in the form of interfaces, scenarios, workflows, and/or look-and-feel assets in order to more realistically validate product hypotheses.

Tactic

Internal Release

Release in-progress development work to allow the product owner to solicit feedback from their organization. Internal releases help evaluate how effectively a product works and may also uncover bugs or performance issues.

Tactic

Product Performance Analysis

Review and analyze product metrics to assess technical performance, user behavior, and business ROI. Determine updates or new features, where applicable.

Tactic

Product Roadmap

Create a guiding document for the development path of a product. Roadmaps can be problem-based or feature-based, depending on the nature of the product.

Tactic

Product Testing

Test product journeys or product features to ensure they are both usable and desirable to the end-user. Activities should be conducted with real users to assess product accessibility, task completion, interface comprehension, and overall fit with user goals.

Tactic

Quality Assurance

Manual and automated activities intended to ensure products satisfy business and user requirements in a systematic, reliable fashion during product development.

Release Planning

Develop a prioritized backlog detailing the design and engineering approach necessary to deliver a validated solution to market. Key considerations for the Delivery track include go-to-market schedule, iteration length, dependency management, and QA and CI pipelines. For the Discovery track, ensure the Delivery backlog is validated through research, data, prototyping and/or user testing.

Tactic

Research Planning

Determine and map out research activities based on the goals of the project. The resulting plan will help align the team and stakeholders on the activities that will effectively generate insights and areas of opportunity.

Tactic

Research Analysis and Synthesis

Review and integrate the results of multiple research streams to uncover insights, opportunities, and critical conclusions.

Tactic

Rolling Release

Release software updates regularly and in short cycles to iteratively improve the capabilities and functionality of an existing product in the market.

Tactic

Solution Architecture

Define system-level components and interactions, including technical requirements and domain boundaries, resulting in a technical vision and plan for execution.

Tactic

Technical Research

Survey and assess technology options, trade-offs, limitations, and possibilities to achieve product outcomes and inform execution feasibility. For existing products, this would include assessing architecture, roadmap, and implementation plans. For new products, this would include reviewing competitive solutions and available or emerging technologies.

Tactic

Technical Validation

Determine whether the proposed concept and/or feature is feasible. This may include mocking up potential solutions and/or integrations to mitigate feasibility risks, validate technical assumptions, and create architectural alignment.

Tactic

Visual Design

Deliver visual design assets iteratively, in tandem with software development cycles. Activities include the creation of interaction models, and visual interface design.

Tactic

User Research

Conduct research to develop a thorough understanding of existing/target user behaviors, attitudes, and needs, plus assess demand. Methods may include generative and/or evaluative research, conducted in order to guide product design and development.

Beta Release

Release of a pre-market version of the product to a select group of users. This allows for the testing and validation of many of the assumptions made during development prior to releasing to market.

Milestone

Insights Shareout

A shareout of key insights based on research synthesis. These will inform the design and development of product concepts and recommend a clear direction on product strategy.

Project Handoff

The transfer of relevant artifacts (e.g. code, designs, research) between teams. Handoffs may occur at the completion of a project or to kick off a new project phase.

Project Kickoff

A meeting between stakeholders and team members to define project and product success and secure overall alignment. Topics covered include business goals, relevant technologies, existing research, and the overall execution plan comprised of select techniques, milestones, and timelines.

Milestone

Product Showcase

An in-depth demonstration of the product in its working-code version, along with a presentation of the technology and overall design philosophy.

Product Strategy Presentation

A formal presentation of a product's strategy that may include the product vision, value proposition, key objectives, measures of success, and/or a design and development roadmap.

Public Release

Release a public-facing version of your product to market, enabling customers to gain value immediately. This is best done when validation and testing have been completed and a wider release has been derisked.

Concept Presentation

A formal presentation of product concepts that address a particular customer profile, business goal, and/or technical space. Concepts may be conveyed using design frameworks, solution maps, interactive features and/or narrative artifacts such as videos.