The Product Thinking Playbook

The Product Thinking Playbook

Product development is difficult. Product thinking is an art and a science, and teams need to rapidly and collaboratively align on a collective product understanding before building.

That's why we developed the Product Thinking Playbook.

Our playbook helps map an execution strategy for new products.

Upon completion of the Playbook exercises, the final “map” becomes your strategic execution plan: a co-created consensus that ensures alignment on purpose and approach.

While not exhaustive, the Product Thinking Playbook helps teams find the right path forward—one crafted to your unique context, capabilities and aspirations.

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 the 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 prescribing the one best answer, but leads to 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 determining 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. 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.

2. 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.

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

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

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

6. Decision points
Should your project need a forking path or a "go/no go" between phases of your plan, insert a Decision Point after the critical milestone.

Index

A cheat sheet of all the tactics, techniques and milestones in the Playbook:

Tactics

Techniques

Dogfooding
Ecosystem mapping
Embedded development
Empathy mapping
Experience mapping
Exploratory testing
External technology exploration
Fake door testing
Feasibility prototype
Field studies
Front-end implementation
Gated rollouts
Gatekeeper collaboration
Goals and metrics setting
Group labs
High-fidelity prototype
Horizon scanning
Hypothesis-driven validation
Ideation
Immersive research
Incident response management
Information architecture
Infrastructure design
Infrastructure development
Insights development
Interaction design
Internal coordination
Internationalization and localization
Kanban
Kano model analysis
Live-data prototype
Low-fidelity prototype
Manual testing
Market sizing
Minimum viable product definition
Monetization strategy
Motion design
Multivariate testing
North star metric
Pair programming
Pairwise comparison
Partner assessment
Performance and error monitoring
Persona development
Pirate metrics analysis
Post-mortem
Pre-mortem
Product analytics audit
Product backlog maintenance
Product branding
Product north star
Product positioning strategy
Quality assurance (QA) requirement analysis
Quality assurance (QA) testing strategy
Qualitative analysis
Quantitative analysis
Release planning
Research materials development
Revenue and cost modeling
Scalability review
Scenario building
Secondary research
Security review
Service blueprint
Software deployment
Stakeholder interviews
Story estimation
Storyboarding
Subject matter expert (SME) interviews
Surveying
Strengths, weaknesses, opportunities and threats (swot) analysis
Systems architecture
Task analysis

Team and project alignment
Team retrospectives
Technical audit
Technical performance review
Technical review and collaboration
Test automation
Test-driven development
Unmoderated testing
Usability testing
User feedback review
User interviews
User recruitment
User segmentation
User story mapping
User story writing
User experience (UX) audit
Value proposition development
Wireframing
Wizard of Oz testing
Working in iterations

Milestones

Alpha / Internal release
Beta release
Concept presentation
Insights shareout
Product showcase
Product strategy presentation
Project handoff
Project kickoff
Public release

Can’t find the right card?

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

Place sticky note here

Can’t find the right card?

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

Place sticky note here

Discovery

Delivery

Tactic

Agile development

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

Tactic

Business research

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

Tactic

Concept evaluation

Evaluate concepts by assessing their potential value to users and the business to prioritize concepts and de-risk directional product decisions. Prioritized concepts will be the focus for further refinement, prototyping and testing.

Tactic

Concept generation

Generate and define multiple product or feature 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, documentation strategy 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 seeks 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

Immersion

Set up a new engagement or phase for success by building team alignment, sharing knowledge of the product/ problem space and setting up tools and processes to embark on the work.

Tactic

Product performance analysis

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

Tactic

Product roadmapping

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 and can include the product vision, north star and minimum viable product definition.

Tactic

Product testing

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

Tactic

Prototyping

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

Tactic

Quality assurance (QA)

Manual and automated activities intended to ensure products satisfy business and user requirements in a systematic, reliable fashion during product development. This may include visual quality assurance (VQA).

Tactic

Release management

Develop a prioritized backlog detailing the design and engineering approach necessary to deliver a validated solution to the market, including relevant documentation. Key considerations for the Delivery track include go-to-market schedule, iteration length, dependency management, quality assurance (QA) and continuous integration (CI) pipelines. For the Discovery track, validate the Delivery backlog through research, data, prototyping and/or user testing.

Tactic

Research analysis and synthesis

Review, sort and create meaning from the results of multiple research activities to uncover insights, opportunities and critical conclusions.

Tactic

Research planning

Devise a research strategy to align team and stakeholders on plans to effectively generate insights and/or areas of opportunity based on the project's goals. Create tactical plans with operational considerations such as budget, timeline and recruitment.

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. This would include assessing architecture, roadmap and implementation plans for existing products. For new products, this would consist of reviewing competitive solutions and available or emerging technologies.

Tactic

Technical validation

Determine whether the proposed concept and/or feature is feasible. Do this by mocking up potential solutions and/or integrations to mitigate feasibility risks, validating technical assumptions and creating architectural alignment.

Tactic

User research

Conduct research to thoroughly understand existing/target user behaviors, attitudes and needs to assess demand. Methods may include generative and/or evaluative research conducted to guide product design and development.

Tactic

UX design

Deploy design techniques to define the user's experience of a product, focusing on the utility, ease of use and delight provided by the product's interactions.

Tactic

Visual design

Create artifacts that define the style and make up the visual language of a product. The goal of visual design within product design is to ensure products are attractive, easy to use and follow best practices for the platform.

Technique

A/B testing

Compare two versions of a product or features to measure which provides the best results. This can include downloads, conversation rate, sales, traffic or any other comparable value metric.

Technique

Accessibility audit

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

Technique

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 support assistive technologies for end-to-end product testing.

Technique

Back-end implementation

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

Technique

Branching strategy

Manage, track and integrate work-in-progess changes to a software product with tradeoffs between integration risk, time-to-release and context isolation. Typical strategies are feature-based branching or trunk-based development.

Technique

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.

Technique

Capability gap assessment

Assess the business in relation to the Service Blueprint and/or Ecosystem Map to determine the capability and strategic gaps, including people, processes, technologies or partners that need to be in place for the desired product experience to be delivered to end users.

Technique

Card sorting

Rank, order and/or cluster a set of features, content, jobs, pains, gains, procedural steps or other items with users, stakeholders or SMEs in order to categorize them and represent their priority or value in relation to a concept, product or experience. Card Sorting can inform information architectures, user journeys and value proposition design.

Technique

Co-creation

Conduct activities to develop concepts or preliminary designs collaboratively with users or stakeholders.

Technique

Competitive review and 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.

Technique

Concept / Feature prioritization

Prioritize concepts or features by evaluating them using a framework or scoring model and an established set of criteria, such as value to users, to the business or technical effort required.

Technique

Concept / Product pitch

Create a pitch to distill and communicate the key elements of the product or concept - the problem, the solution, how it works, who it benefits, why it counts and a call to action. Product Pitches use storytelling to engage and enable decision-makers and/or customers.

Technique

Concept / Product video

Craft a promotional or demonstrative video that showcases a product or concept. Use storytelling to illustrate the value proposition while highlighting key features.

Technique

Concept development

Mature product ideas or concepts to create a more refined understanding of the value proposition, user experience, technology involved, potential features, risks and enablers. Outputs may take the form of concept briefs, posters, storyboards or videos.

Technique

Concept testing

Present the product or feature concepts to potential users to get preliminary feedback on their value and desirability. Use this as an opportunity to ideate, refine and improve aspects of the experience.

Technique

Continuous integration / Continuous delivery

Develop software with automated processes that ensure successful integration and repeatable automated building, packaging and deployment.

Technique

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.

Technique

Customer profile development

Clarify understanding of user segments by mapping their jobs, pains and gains into a Customer Profile Canvas (CPC). Ensure jobs, pains and gains align to a particular product experience or opportunity area, as informed by research. Do further research/testing to validate all data points.

Technique

Daily standups

Daily meetings or synchronization around what a product team is doing at the individual level and any risks or blockers to success.

Technique

Data architecture

Define the systems, schemas or patterns of data handling and storage. Ensure consideration for scalability, security, reliability, performance and model simplicity or compatibility.

Technique

Data audit

Perform a deep dive exploration of internal and external data sets (including their size, relevance, availability and suitability) for a use case or as an input to Data Architecture.

Technique

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.

Technique

Defect management planning

Design and implement processes to efficiently coordinate defect detection, root cause analysis, defect triage and reporting across product team(s) and stakeholders. Effective defect management incorporates both the processing of defects and interplay with Agile Rituals to address systemic sources.

Technique

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.

Technique

Design system

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

Technique

Diary studies

Collect qualitative information over a set period of time by having participants record entries about their everyday lives or experience with a product in either written, survey or video form.

Technique

Digital ethnography

Observe natural social interactions within online communities to understand attitudes and behaviors regarding a specific topic, product or service.

Technique

Discovery / Delivery sprint 0

Ensure readiness for future product strategy, research, design and/or agile development sprints. Consume existing problem/product research, analytics, requirements or backlog and develop a Northstar to align progress. Typically includes setting up team processes, tooling and a minimal development environment.

Technique

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.

Technique

Ecosystem mapping

Map the key actors/roles that relate to, or influence a product, service or business, to create a representation and understanding of the surrounding ecosystem.

Technique

Embedded development

Implement software that runs on a specialized edge device with considerations for integration with sensors and input/output (I/O) integration.

Technique

Empathy mapping

Map what is collectively known about a type of user to create a shared understanding of their emotions, attitudes and behaviors.

Technique

Experience mapping

Map a user's journey or experience of a product to identify needs and opportunities. Then, map the steps or actions of each stage of the experience and any relevant factors, including actors involved, tools or technology used, emotions, pain points and more.

Technique

Exploratory testing

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

Technique

External technology exploration

Determine what interfaces, such as Application Programming Interfaces (APIs) and Software Development Kits (SDKs), components, such as libraries, and development environments or frameworks are available from a particular technology or for a particular use case. Inspect the feature set and test capabilities; document and communicate findings.

Technique

Fake door testing

Provide users with an option that does not yet exist and measure engagement to validate demand rapidly. Fake doors could be an advertisement, button, pop-up or another interactive element that suggests the existence of a product or new feature.

Technique

Feasibility prototype

Build a technical prototype with just enough code to test new technologies (e.g., algorithms, hardware, specific features, functional areas) to address technical feasibility risks, often related to performance, during product discovery.

Technique

Field studies

Observe and interact with participants while they are in a natural environment to make observations first-hand and in context. Activities may include direct observation, contextual inquiry, limited participation and document analysis.

Technique

Front-end implementation

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

Technique

Gated rollouts

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

Technique

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.

Technique

Goals and metrics setting

Identify and align on clear product and business goals and outcomes, and how to measure if the product is successful. May be done with the creation of a product 1-pager, objectives and key results (OKRs), success metrics or goal laddering.

Technique

Group labs

Collect data from representatives of a target audience through group interaction. Lead a group of participants through discussion and/or activities relating to a given topic to reveal and explore commonalities and differences among them.

Technique

High-fidelity prototype

Create a prototype that resembles the final product or feature in terms of functionality and design details. High-fidelity prototypes can evaluate a product's user experience, from visual design to interaction.

Technique

Horizon scanning

Detect early signals of change in core and peripheral environments using foresight frameworks to uncover new modes of thinking, provoke solutions from alternative perspectives and develop strategies for the mid-to-long term.

Technique

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 product's success and/or their relative risk) and identify appropriate methods to validate or invalidate these assumptions one by one.

Technique

Ideation

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

Technique

Immersive research

Experience the journey of a product or service first hand to investigate and document the process, and to better understand the user experience.

Technique

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. The resolution of incidents is a combination of defect management and/or software deployment.

Technique

Information architecture

Create, organize, structure and label information-rich systems that support discoverability, findability and task completion for users. Activities can include content auditing, taxonomy development, information grouping, card sorting, site mapping and more.

Technique

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 factors. Activities may include prototyping and evaluating alternatives for data ingestion, storage and processing, which can enable impactful artificial intelligence (AI), machine learning (ML) and data science outcomes.

Technique

Infrastructure development

Implement components on which business logic or data handling is built.

Technique

Insights development

Distill insightful learnings from research activities and articulate them into consumable and actionable outputs. Insights should explain both the ‘why’ and the ‘so what’ of user behavior to guide a path for better decision-making.

Technique

Interaction design

Design the interactions that will enable users to achieve their objective(s) within a product. Interaction design considers the words, visuals, objects or space, time and behaviors of each interaction.

Technique

Internal coordination

Cross-functional coordination, training and mobilization of product, sales, marketing, distribution, customer service and compliance teams required to deliver the product to market successfully.

Technique

Internationalization and localization

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

Technique

Kanban

Continuously develop software by maintaining a prioritized backlog of tasks or stories and managing flow between stages of development to reduce work in progress at any stage.

Technique

Kano model analysis

Use customer data to help you prioritize your backlog of features. This product framework categorizes features into five categories according to the degree to which they are likely to satisfy customers.

Technique

Live-data prototype

Build a coded prototype that uses real data and live traffic to stress test a product concept in real-world conditions. Live-data prototypes require limited implementation — typically just the critical use cases and none of the "productization" — to prove something works.

Technique

Low-fidelity prototype

Create a quick, minimal prototype to translate a high-level product concept into a testable artifact (paper sketch, storyboard, wireframes). The most important role of lo-fi prototypes is to assess the value proposition rather than the product's design.

Technique

Manual testing

Leverage product knowledge and structured scenarios to manually perform the initial verification of new or updated functionality. This is suitable during Exploratory, Usability and/or ad-hoc testing for non-repetitive tests that require human observation.

Technique

Market sizing

Estimate the maximum size of the opportunity for a particular product or service (Total Addressable Market) and how much the business might earn from it (Serviceable Addressable Market and Share of Market). Market sizing can be used to evaluate potential opportunities and de-risk early investment.

Technique

Minimum viable product definition

Define a new product's minimal feature set that reflects its core user value proposition. Ensure it's aligned with validated and prioritized product goals, features, dependencies and constraints. Release to early adopters, generating lean feedback from the market.

Technique

Monetization strategy

Plan how a product or service will generate revenue from the value it delivers to its users. Consider the product's business and lifecycle to determine the appropriate monetization mechanism.

Technique

Motion design

Apply motion design techniques, such as animating interactions, navigation or transitions, to enhance the usability and delight of a product.

Technique

Multivariate testing

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

Technique

North star metric

Align on the key measure for success for product development teams in a company. The North Star Metric helps teams align over a unified story for success and a leading indicator of progress.

Technique

Pair programming

Develop software as part of a pair of software engineers sharing context but dividing responsibility for strategy or structure and implementation. Pair programming can expand to include mobbing or other ensembles.

Technique

Pairwise comparison

Evaluate options (i.e., features, concepts) by having participants or teams compare them within sets of pairs. Have participants or teams decide on a preferred option. A pairwise matrix is then used to determine a ranking.

Technique

Partner assessment

Explore and assess potential partners or suppliers to acquire resources, capabilities or access and reduce risk in delivering a product or service.

Technique

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 and others.

Technique

Persona development

Create one or multiple personas that serve as fictional representations of a specific type of user to understand key traits, behaviors, emotions, goals, responsibilities and needs.

Technique

Pirate metrics analysis

Examine data that measures how users navigate a product to reach a desired conversion point. Pirate Metrics Analysis uncovers where users drop off and prioritizes where to dedicate efforts to improve conversion rates.

Technique

Post-mortem

Conduct a Post-Mortem exercise after the completion of a project to celebrate wins, reflect on the team's workflow and iterate on any processes to make things better for next time.

Technique

Pre-mortem

Conduct a Pre-Mortem exercise ahead of a project or big release to identify product and process risks, assumptions and dependencies that could lead to failure and look for ways to mitigate them.

Technique

Product analytics audit

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

Technique

Product backlog maintenance

Create and iteratively maintain a single, authoritative list of user stories, tasks and/or defects, which may be effort estimated and sequenced into sprints, that reflects an evolving list of items that a team works on to make progress toward a target outcome.

Technique

Product branding

Develop and apply a brand strategy to the design of a product that reflects the appropriate tone and voice. Outputs may include a product name, logo, graphic elements, user experience (UX) writing, marketing materials and more.

Technique

Product north star

A living document that connects the product's vision to specific and measurable business and user goals realized through incremental product feature releases. Continuous research into the user's job, pains and gains, and feedback on the product's value to the business and its users inform this artifact.

Technique

Product positioning strategy

Define where the product or service fits in the marketplace, why it's better than alternative solutions and why customers should care about it. The goal is to communicate how the product directly and uniquely serves customers' needs.

Technique

Quality assurance (QA) requirement analysis

Beginning with a validated solution backlog, quality assurance (QA) analyzes product requirements, design specifications and system interactions to validate accuracy, comprehensiveness and testability.

Technique

Quality assurance (QA) testing strategy

Define the test-execution approach and plan that outlines the test scope, objectives and processes to ensure product quality and quality assurance (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.

Technique

Qualitative analysis

Organize research observations, evidence and/or ideas into thematic groups to uncover patterns via coding and tagging, affinity mapping and/or using a research database.

Technique

Quantitative analysis

Transform raw numbers into meaningful insights by identifying relevant data sets, cleaning data, running calculations or modeling, examining relationships and trends in data and disseminating relevant findings.

Technique

Release planning

Determine the target outcome(s) of one or more major releases and establish a plan that identifies release or product goals, target features or functionality, key partners, schedule and budget constraints, critical dependencies and how progress will be measured.

Technique

Research materials development

Prepare research planning documents, facilitation guides and stimuli needed for research. Use these materials to define research objectives and guide activities to create tactical plans for research execution.

Technique

Revenue and cost modeling

Lightweight forecasting of potential revenues and profits of higher fidelity concepts or working products using informed viability assumptions to determine volume, price and cost inputs.

Technique

Scalability review

Review a software product for limitations on scaling. Considerations should include queueing, asynchronous process, reactive programming, horizontal scaling and data locking.

Technique

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.

Technique

Secondary research

Access existing data from reliable sources to support the exploration, understanding and synthesis of topics of interest during research.

Technique

Security review

Review a software product for security. Use techniques such as code review, dependency audits, static analysis and penetration testing to assess security risks and potential vulnerabilities.

Technique

Service blueprint

Map the user experience of a product along with all the business operations needed to deliver the intended experience. Pay attention to “frontstage” touchpoints that the user sees and the “backstage” people, processes and technology that are largely invisible.

Technique

Software deployment

Deploy a software product to end users by releasing software and enabling features. Software Deployment typically follows an expanding scale with internal releases, preview or limited releases and general availability. In addition, software deployment may include rollback, downgrade, reset or recovery techniques.

Technique

Stakeholder interviews

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

Technique

Story estimation

Estimate the effort or complexity of a backlog item as an input to sprint planning and velocity.

Technique

Storyboarding

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

Technique

Subject matter expert (SME) interviews

Interview individuals with knowledge of a specific subject to gain a deeper understanding of it. These can be used at any stage of product development but are commonly conducted during Immersion to build context, and during Research to understand complex or niche subject areas.

Technique

Surveying

Issue a questionnaire to a targeted group to identify common problems, behavioral patterns and perceptions. Surveys can also validate proposed product directions relating to desirability, usability or viability.

Technique

Strengths, weaknesses, opportunities and threats (SWOT) analysis

Identify a product, service, or business's strengths, weaknesses, opportunities and threats (SWOT) relative to the market for strategic planning and validation.

Technique

Systems architecture

Define the interaction of software components and services using common design patterns to trade off isolation, maintainability, scalability and extensibility.

Technique

Task analysis

Study how people complete tasks in order to understand their motivations or method of execution. Task analysis may be used to understand existing behaviors or in order to evaluate how users attempt to complete a task using a product or prototype.

Technique

Team and project alignment

Ensure the team is equipped to work together including team forming and norming, RACI (responsibility, accountability, consulted, informed), and collaboration and communication models. Align teams on the product/project goals including project plan and north star, product visioning and problem framing.

Technique

Team retrospectives

Regularly examine the good, bad and unclear of a product team's work, environment and interaction. Retrospectives should produce action items for continuous improvement.

Technique

Technical audit

Perform a deep dive evaluation of current software, tech stack and architecture for scalability, security, performance or maintainability recommendations.

Technique

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.

Technique

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, a deeper reflection of implementation approaches, knowledge transfer across technical domains and shared code ownership across the team.

Technique

Test automation

Verify existing functionality and behavior in a consistent, repeatable fashion. Test Automation moves the responsibility of regression testing to code, enabling faster release cycles and a better distribution of testing tasks.

Technique

Test-driven development

Use a technique to define the target behavior as a failing unit test, then create the necessary logic to pass the test. Typically this is implemented in a red/green refactor methodology, which espouses simple changes followed by clean refactoring.

Technique

Unmoderated testing

Asynchronously test a product's or prototype's usability and comprehension by recording participants' interactions without real-time observation. Use tasks to guide their independent exploration, and embed questions to prompt feedback.

Technique

Usability testing

Evaluate an interface's efficiency, effectiveness and enjoyability 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.

Technique

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 center data and app store reviews.

Technique

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.

Technique

User recruitment

Recruit participants for research activities by defining, locating, inviting and scheduling appropriate representatives of your target audience into your user research. This may involve engaging a third-party recruitment agency or tool.

Technique

User segmentation

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

Technique

User story mapping

Map a product's user goals sequentially to articulate the activities and tasks they might perform to help them achieve their goals. This will help define and prioritize product features for minimum viable product (MVP) releases and beyond.

Technique

User story writing

Create actionable summary descriptions about the features of a software system using natural language. In Behavior Driven Development, user stories have two parts: the narrative, which defines the user or business value of a specific feature part, and the expected behavior detailed as acceptance criteria.

Technique

User experience (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. This is often an informal practice but can materialize as a formal document if required for knowledge transfer.

Technique

Value proposition development

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

Technique

Wireframing

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

Technique

Wizard of Oz testing

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

Technique

Working in iterations

Plan and complete an iteration to build a shippable unit of a software product. Iterations are time-boxed with selected input tasks or stories and concluded with a review or demo.

Milestone

Alpha / Internal release

Release in-progress development work to internal employees and/or friends and family to allow the product owner to solicit initial feedback. Alpha releases help evaluate a product's effectiveness and may also uncover bugs or performance issues.

Milestone

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

Concept presentation

Present product concepts that address a particular customer profile, business goal and/or technology space. Convey concepts using design frameworks, solution maps, interactive features and/or narrative artifacts such as videos.

Milestone

Insights shareout

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

Milestone

Product showcase

Demonstrate a product in-depth to exhibit a working code version that accurately showcases the product's technology and overall design philosophy.

Milestone

Product strategy presentation

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

Milestone

Project handoff

Transfer all relevant artifacts (e.g., code, designs, research) and documentation between teams. Handoffs may occur after project completion or to kick off a new project phase.

Milestone

Project kickoff

Meet with 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

Public release

Release a public-facing version of your product to market, enabling customers to gain value immediately. This is best when validation and testing are already complete, allowing for a wider release with mitigated risk.

Decision point

Decision point

Decision point

Decision point