The Cylinder Way

How We Grow Your Business by Building Internal Startups inside Your Company

V1.2.3 - Last edited March 21, 2018 by Adarsh Pandit

Hello!

We are very glad to meet you and get to know more about your business.

This document describes how we help partners grow their businesses by building internal startups. We bring startup-quality design, agility, and speed to large companies, helping build new internal incubators and accelerating existing startup initiatives.

We help partners reach their business goals using human-centered design and rapid software development.

This document describes our work, our process, and our company. The highlights of this document include:

- Typical Project Types
- Estimating a Budget
- A Typical Project, Typical Week, and Typical Day
- Our Values
- Next Steps

Why You're Here

You have a business. You want to grow revenues or cut costs. You believe software could help you accomplish your goals.

Where You Want to Go

You want to grow your core business, expand into an adjacent business with a new product/service, or you have streamlined an internal process. You might need a new marketing site or expert engineers.

How We Can Help You

We design and build custom web and mobile applications to help businesses achieve their goals. If an off-the-shelf solution fits your needs, we will strongly recommend it over custom software.

1. We start by understanding the problem through structured research.
2. Then, we quickly build lightweight prototypes and test them with real people.
3. Finally, we build on the best prototype and continuously refine it through user feedback and research to ensure a successful solution.

We will communicate frequently and clearly. We will adjust our process to fit your business and context. We will help transition the solution to full-time staff once implemented. We address and resolve ambiguity. We do our best to make our time together hassle- and headache-free.

We will make you look good in front of your boss and let you take the credit.

Getting Started

Before We Begin

Introductions

Somehow, we met one another, maybe through a mutual friend. Now it's time to learn about one another's businesses and see if we should work together.

Some questions we might ask at this point:

- What are your values? What is important to you?
- What is the history of the business? This project?
- What are the a) areas for growth and b) operations to be streamlined?
- What are your concerns?

Research Workshop

We start projects with a one-week Research Workshop. We spend this time investigating the problem, drafting potential solutions, aligning the team, and producing a detailed project estimate/timeline.

The main outputs of the workshop are **Research**, **Alignment**, and an **Estimated Budget**.

Deliverables

Research

Inspired by Google Venture's [Design Sprints](#), we start a project by understanding the problem and prototyping potential solutions. Everyone gets together in a conference room for a week, including the main client decider. We fill the room with healthy snacks, water, post-it notes, and colorful markers.

To start, we work to understand the context of the business goal, team, company, and industry in a short amount of time. We may interview the client team, fill out a Business Model Canvas, do snap analysis of competitive products, identify hypotheses, and capture assumptions.

We then diverge and generate lots of ideas for solving the problem using markers, paper, and white boards. Taking those ideas, we converge on the better ones, and prototype/test those ideas with actual people.

The week is very structured, but flexible and is designed to adapt to the specific client context. We do this work to help the team surface knowledge, empathize with the user, and generate potential solutions quickly.

Whatever we end up doing, we'll compile all of the learnings in a specification paper. This document is the deliverable for the end of the phase.

Alignment

Everyone comes into a new project with different expectations. Projects are most successful when expectations are collapsed into a single team vision before starting any actual work. We accomplish this by talking as a group.

We want to align on:

- The purpose of the project and how it directly helps the business grow revenue or cut costs.
- A User Persona. Who is this software for and what goals are we helping them achieve?
- A User Journey Map or what we expect the persona's experience to be with our software.
- Assumptions and hypotheses.
- Who the important people are, from the working team to the executive sponsors, as well as what everyone's roles are.
- Communication protocol, including meetings, tools, and schedules.
- Design and development protocol, including software design and development workflows, best practices, and terminology.
- The biggest risks to the project. Imagine this project has failed. Why?
- Milestones, budget, and timelines.

Estimated Budgets

> *"I'm going to embrace the fact that I'll know more about your project tomorrow than I do today." - Rob Harr*

The great struggle of software design and development is the need to know what we're building before we can know how much it may cost to build. This is why we start with the Research Workshop.

We intentionally use the term "budget" because each project is an investment with a cost and expected return. Our goal is to help you decide on how much to invest in your project

and give feedback on how reasonable it is, or what might get accomplished within that budget.

The three variables for software project management are budget, time, and features. Two of these can stay fixed, one has to be variable - in industry jargon, we call this [The Iron Triangle](). We do our best work when the budget and timeline are fixed. The scope is then flexible for adapting to new information and ongoing user research.

How We Agree on a Budget:

- Define high-level goals
- Break those down into a series of weekly themes. These should be a picture of the application at the end of each working week.
- Identify necessary infrastructure setup or onboarding time at the beginning
- Identify "fit and polish" time at the end
- Schedule regular user testing

The Cylinder System of Software Design and Development

Over the years, we have developed a specific process for building software which keeps costs down, allows teams to move quickly, and build the most useful software for our users.

We do this by focusing on high-level planning for the short term, short feedback loops, automating best practices, high-quality communication, and continuous improvement of our process.

Our Way is a form of [agile software development](), drawing from [Kanban]() and [The Agile Manifesto]().

It is the best way we know to build great products for users and have happy clients.

Guiding Principles of Cylinder Development

- New features start from conversations with users
- Everyone works toward a few big weekly goals set by the Product Owner
- Focus on clear, high-level, short-term plans
- Continuous daily planning and reprioritization
- Continuous communication and problem-solving
- Everyone owns the project management process, but one person leads it.
- Simple, low-feature project management tools like [Trello]() or yellow stickies

Team

The team will include Designers, Developers, and Partners.

The Designers and Developers (AKA The Team) will be the main points of contact for ongoing project-related discussion / planning. They work on one project at a time, all week.

Designers

Cylinder Designers are multi-disciplinary experts who are responsible for ensuring user needs are being met, and creating a clear path to completing the [Job to be Done]().

Their expertise includes:

- Product Design (prototyping, facilitating Design Sprints, sketching, strategic direction)

- User Research (organizing interviews and user-testing prototypes and software)
- Visual Design
- Front End Development

It's a lot of things to be good at. We staff designers in pairs, where they can trade off wearing the above hats. It's a lot of things for one person to be responsible for.

Throughout the project, designers are responsible for reminding us that we are writing code for users and serving as an advocate for those users.

Developers

Developers are responsible for writing high-quality, well-tested code. They are also excellent consultants and can help with planning, project management, estimation, prioritization, tool choice, and feedback.

All members of the Cylinder team are highly technical and very experienced in Ruby/Rails/JavaScript/SQL. All are capable of working on code, design, and informing product decisions as well as technical architecture design. Both the designers and developers will provide input and suggestions when making UX decisions, engineering decisions, and forming a roll-out plan.

Partners

The Partner is the contact for questions about invoicing, staffing and scheduling, and as a high-level advisor. They work on many projects at a time, but make themselves available on demand.

Communication

- Cylinder team members will provide updates and be available via secure channels such as Slack, not email.
- We will establish a persistent video channel for meetings and virtual co-working.
- We will schedule in-person brainstorming and strategy sessions as needed.

Typical Projects

We have some typical project types and have described them in some detail below. Please skip to the relevant section for your business. These are meant to be guidelines, not strict service packages. [Contact us to discuss further](#).

1. [Product Design and Development](#)
2. [Expert Engineering Support](#)
3. [Marketing Site Design](#)
4. [Coaching / General Consultation](#)

Product Design and Development

Overview

When a company's business goals are best advanced by designing and building a digital product, this is the project type which fits best.

Example business contexts where this project type works well includes:

- "We have a great current business and would like to break into a new adjacent space"
- "We have a labor-intensive internal process and would like to automate parts of it."

The budget for a project like this is typically $100-300k over several months.

Build Phase

After [the research workshop](#), we begin writing software. This is the bulk of the time we will spend together, building and iterating on the project. Details are in [A Typical Week](#).

Handoff Period

With 3-4 weeks left in the project, we will plan for transitioning this project back to you and your development team. Your developers and designers will lead meetings and take over the project while we play a support role. Ideally, your team has been involved in the project since the beginning, but we know that's not always an option.

If you do not have a team, we're happy to do maintenance work on a long-term basis or help you hire a team. Typically we do this kind of work as a monthly retainer. For more information, write to sales@cylinder.work.

The Cylinder Way - cylinder.digital

Fit and Polish

We prefer to leave 1-2 weeks at the end of the project for final testing, proofreading copy, and high-level review. It's important to design this into our plan from the beginning. It's not a lot of work but can make a big difference in first impressions. Also, this is most impactful at the end of the project, not along the way.

The launch date should be set well before this phase.

Celebration!

We should get together and celebrate our successes!

A Typical Week

Our projects are planned on a week-to-week basis, with a consistent eye on the larger timeline and project vision. Here's how that typically looks.

Monday Planning Meeting

At the start of each week, we spend about an hour planning our work. Our goals for the meeting are:

- Ensure we have enough to work on for the next few days
- Walk through each of the upcoming work and ensure the descriptions are clear
- Break down large features into smaller ones
- Order the features from most important to least important
- Revisit the overall project plan and goals. How far along are we? Does the end goal still seem reasonable?
- Plan user testing
- Discuss our shipping cadence, and anything slowing that down.

User Research

At a regular cadence (every ~2 weeks), we engage in User Testing of our application. This is a structured scientific interview of potential users where we show our live application in order to get feedback.

After this research, the designers will synthesize the research results and write new features for some proposed changes to the application. The designers will review these changes with the client lead and agree on the new prioritization of work.

Ideally, we do this later in the week, so we have adequate information for the Monday planning meeting.

Retro and Demo

At the end of the week, typically on Friday, we conduct a Retrospective Meeting.

We start the meeting with a group demonstration of the current state of the application, to center the team on the actual work output and celebrate how much has been accomplished. While we may make notes for small bugs, this is meant to be a milestone marker, not a design review.

The retrospective takes up the majority of the meeting. The main goal is to reflect on the processes and systems which the team is using, figure out what's working or not, and propose changes.

There are lots of ways to run a retrospective, but here's the one we propose:

- Open a spreadsheet with three columns: happy emoji, meh emoji, sad emoji. Also stickies.io works well.
- Everyone fills in the columns with topics for discussion. They do not have to be evenly distributed.
- Anyone can ask for an item to be "starred" for further discussion.
- The first half of the time is this brainstorming activity. The second half of the meeting is a deep dive on the starred items.
- Facilitator collects "todos" as we go and is responsible for following up.

A Typical Day

We build in brief daily touch-points to keep everyone aligned and on track, and maximize the solo/focused work time for peak productivity. Here's what that looks like.

Video Standup

Every morning at the same time, the whole team spends about 30 minutes together on a video call having a quick catch-up meeting. It's a time to get together and plan our upcoming day. We have a round-robin discussion where we take turns sharing what we worked on yesterday and plan for today.

This is a great time to ask for help getting unblocked or coordinating with someone else to pair on a task. It's also really helpful for hearing about non-work topics to build stronger interpersonal relationships and trust.

Continuous Planning

[Cylinder Software Development](#) is the practice of continuous, short-term, high-level planning and execution, so standup is a great regular time to quickly review the plan. During the standup, the Project Lead will review the upcoming day's work with the team, feature by feature. Teammates can ask clarifying questions, break larger work into smaller pieces, suggest a different prioritization, remove stale or obsolete stories, or propose new ones.

Everyone owns the planning process but one person leads it. As the requirements or context for the project change throughout the day, the Project Lead will adjust the upcoming queue of work.

Daily Communication

Throughout the day the team will communicate in channels such as Slack (chat) and ad hoc Video Calls. Email is to be used very rarely.

Continuous Delivery

The units of software change (typically a GitHub Pull Request) should be small and focus on one area of the application or theme. The code is then reviewed by a teammate. Small units make for fast reviews and happy reviewers.

After review, the code is pushed to a non-production staging environment for testing. If I worked on the feature, my judgement is a little cloudy and someone else should ensure the app is behaving in the expected way, then approves the change.

Finally, the code is then deployed to production. Deploys happen several times throughout the day.

The production server will always have bug-free working software, however minimal. Small, frequent iterations reduce risk and allow us to move extremely fast.

Expert Engineering Support

Overview

When a company's business goals are best advanced by adding expert engineers to an existing team, this project type is suitable.

Example business contexts where this project type works well includes:

- "We have an urgent standalone project with clear success metrics but are engineering constrained."
- "We have the product design skills but lack the team to deliver the software."
- "We have a great team who could use some guidance and training from an expert."

Note: "We just need more bodies" is an anti-pattern for us and likely we are not the best fit for this kind of project.

The budget for a project like this varies greatly, depending on the length of support and size of the team, but can be $50-400k.

Typical Week

We will work within your existing software design and development process.

If you are open to process recommendations based on our years of experience and untold number of client projects, we would be happy to work with you.

If it were up to us, the typical week would look like the one we described above.

Marketing Site Design

Overview

When a company's business goals are best advanced by presenting a better marketing landing page, this project type may be most appropriate.

Example business contexts where this project type works well includes:

- "Our landing page doesn't convey what we are selling to customers."
- "Our landing page does not convert visitors to customers well."
- "Our branding is inconsistent with our values or goals."
- "Our landing page is not presentable or out of date."

The budget for a project like this varies based on the number design feedback iterations, but can be $15-50k.

Coaching / General Consultation

Overview

When a company's business goals are best advanced by having an executive expert guide the operations with a light-touch, this may be the best project type.

Example business contexts where this project type works well includes:

- "Our software development process is frustrating for our team."
- "We are not shipping features and iterating as quickly as we should."
- "We are having a hard time hiring great teammates."
- "We need help working on our business model or understanding the risks to our business."

The budget for a project like this varies based on the time commitment per week and length of project, but can be $10-50k.

Our Firm

Company Values

- We work to live, not the other way around. Friends and Family First.
- We only work on things we are proud to put our name on.
- Organizational change should also be prototyped - propose small changes, try things out, and evaluate how it works as a group.
- Start with the end in mind. Plan toward that goal.
- Software is a human-centric process. People build software for other people to use.
- Communication is the cause of, and solution to, all of our work problems.
- Invest in business relationships, which take a long time to build. Trust is slowly built and easily lost.
- An inclusive team builds better software.
- Shipped is better than perfect. Deploy every day, all day.
- All of our tools are open-source - we need to contribute back to pay down the debt.
- Start somewhere immediately and iterate frequently
- There are no opinions, only hypotheses in need of testing
- Small teams are better than big teams.
- A few high-level goals are better than lots of detailed plans.
- Implementers should have a lot of autonomy. They are experts.
- Time is precious. We should not spend it commuting to work unless we decide doing so makes the best sense.

Working Norms

- We have a distributed team intentionally. This takes work to get right and it works well for us. Read here for more.
- We keep regular working hours, Monday-Thursday or Friday depending on your personal needs. No weekends, no all-nighters.
- After a fire drill, we should pause and ask what happened and how we can prevent this kind of urgency in the future.
- Our company should reflect the world we want to live in. If your work is disrupted by extreme weather, we support paid time off as you cope. Read more here.

Invoicing, Contracts, and Payment

Invoicing

Our projects are billed on a time-and-materials basis only. [We do not accept fixed-bid projects](#).

We send invoices weekly on Saturday. They are due on receipt.

Contracts

Our contracts are pretty simple. The salient points are:

- Invoices sent weekly, terms "due on receipt". If enough invoices go unpaid, we suspend the project. This has yet to happen with a Cylinder client.
- Either party can terminate the agreement with 2 week's notice (10 working days) for any reason. Frequent communication between project leads typically makes this clause irrelevant, but it's a good protection for everyone.
- The client owns all code and intellectual property (AKA "work for hire").
- Everything is confidential unless you say otherwise in writing.
- If you page us outside working hours, we will consider that "emergency time" and it's billed at a higher rate. This has to either be explicitly invoked by you in writing (like an email) or have an agreed upon rule in place which automatically treats some condition as an emergency.
- Travel: we can come to your offices for 3-4 days of the first week, then as-necessary. The travel time and expenses are billable.
- Two weeks pre-payment is required to initiate work.

No Fixed Bids

Some consulting relationships start with a requirements document or RFP ("Request For Proposal"). The requirements are often extremely detailed.

The probability of this document containing the optimum feature set is extremely low. The right features are better learned through user interviews, prototyping, releasing actual software, and getting feedback from real users.

Based on that document, clients expect consultants in the industry to submit an exact timeframe and bid. This contract style sets the client and consultant working against each other right from day one. Instead of focusing on designing the product experience or evaluating what assumptions were wrong, they spend time negotiating about what was

The Cylinder Way - cylinder.digital

meant in a document written a long time ago or focusing on arbitrary deadlines. But it's worse than negotiating; it's retroactively discussing something that no one remembers the same way.

As you might have guessed, we don't do fixed-bid, fixed-feature-set proposals.

Payments

We prefer wire transfers into our bank account but can accommodate other methods if you can cover the fee. Each invoice has payment details in the "notes" section. If you have questions, write to accounting@cylinder.work.

Next Steps

Our primary goal is to help our partners with business-critical projects which help them grow and deliver amazing products to their customers.

To start this project, we need a deposit of two week's fees in the bank before we have a kick-off meeting.

We would need signed contracts. We're happy to share ours with you ASAP - legal review is usually longer than our discussions take.