

#Dev Different

Why Software Is Hard
& How To Fix It

This workbook prepared for: Elev
Issued: 0.24

CodeBuddy
Omaha, Lincoln, Austin
devon@codebuddy.com

"You get good product from having a good company. You don't get a good company from having a good product."

Devon Seacrest

About The Speaker

#DevDifferent

#DevDifferent



Devon Seacrest, co-founder and CEO of CodeBuddy, is a repeat co-founder with over 24+ years of owning and operating high growth businesses. Devon has raised over \$27 million dollars in software and real estate capital and was a 2016 Kauffman Foundation Pipeline Fellow. He is a full-stack software engineer, university professor, graduate of UNL College of Law, and bilingual (Spanish).

CodeBuddy is a team of technical founders with a background in creating systems intelligence and management software for federal law enforcement and Department of Defense.

With over two decades of experience in start-ups and enterprise software, I have seen many projects run millions of dollars over budget and years longer than scheduled, only to be put on the shelf.

In the search for a better and faster way to build new software, I've met with hundreds of entrepreneurs to discover the tools, beliefs, and processes they use to successfully disrupt their competition. This workbook shares those secrets.

A stylized handwritten signature in black ink, consisting of a series of loops and a long horizontal stroke.

Devon Seacrest



"You get good product from having a good company. You don't get a good company from having a good product."

Preview

#Dev Different

#DevDifferent

Preview



Chapter 1: Future of Software

Future trends in software development and use.



Chapter 2: Why Software Is Hard

Why software is always over time and budget.



Chapter 3: #DevDifferent

Tools, processes, and techniques to make to build better software faster.



Chapter 4: CodeBuddy rIDE

A tool that designed to help less technical and technical founders build software.



Chapter 5: Question & Answer

Other frequently asked questions and answers.

Why Read This Workbook?

We've seen too many companies go 200% over time and over budget on software projects that end up getting shelved. Read this to avoid their mistakes.

"Progress happens when it takes less time and resources to do the same task you did yesterday."

Development Velocity and Customized Software

Future of Software

#DevDifferent

Future Of Software

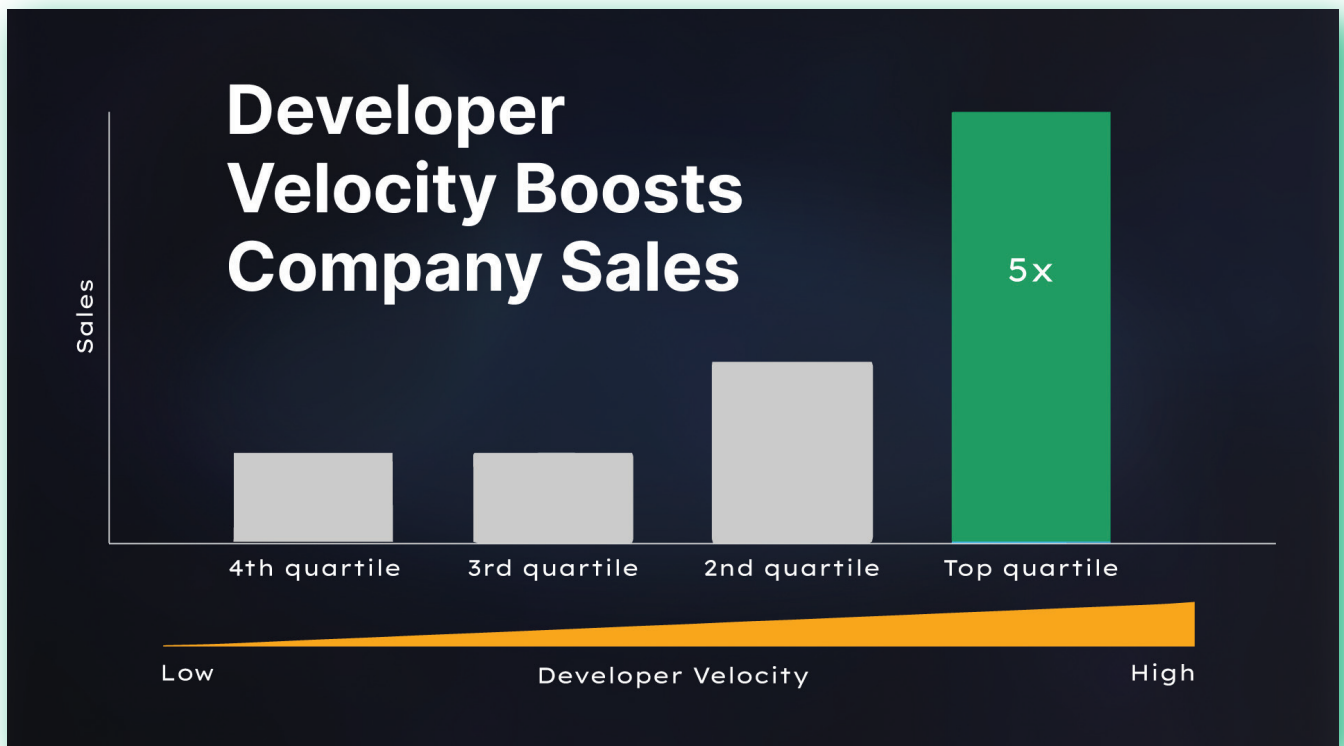
We see two main things shaping the future of software.

Development Velocity Is Critical

Organizations that invest in faster developer velocity have five times the revenue of their competitors. They also have 60 percent higher total shareholder returns, 20 percent higher operating margins, and perform better on customer satisfaction and brand perception.

AI Allows for More Customized Software

The advantages of AI and software development tools like CodeBuddy are allowing companies and individuals to build software that they could have never previously offered faster than they ever could have imagined.



Current Trends

Seeing \$200K apps built for \$5K in a matter of weeks.

"Progress happens when it takes less time and resources to do the same task you did yesterday."

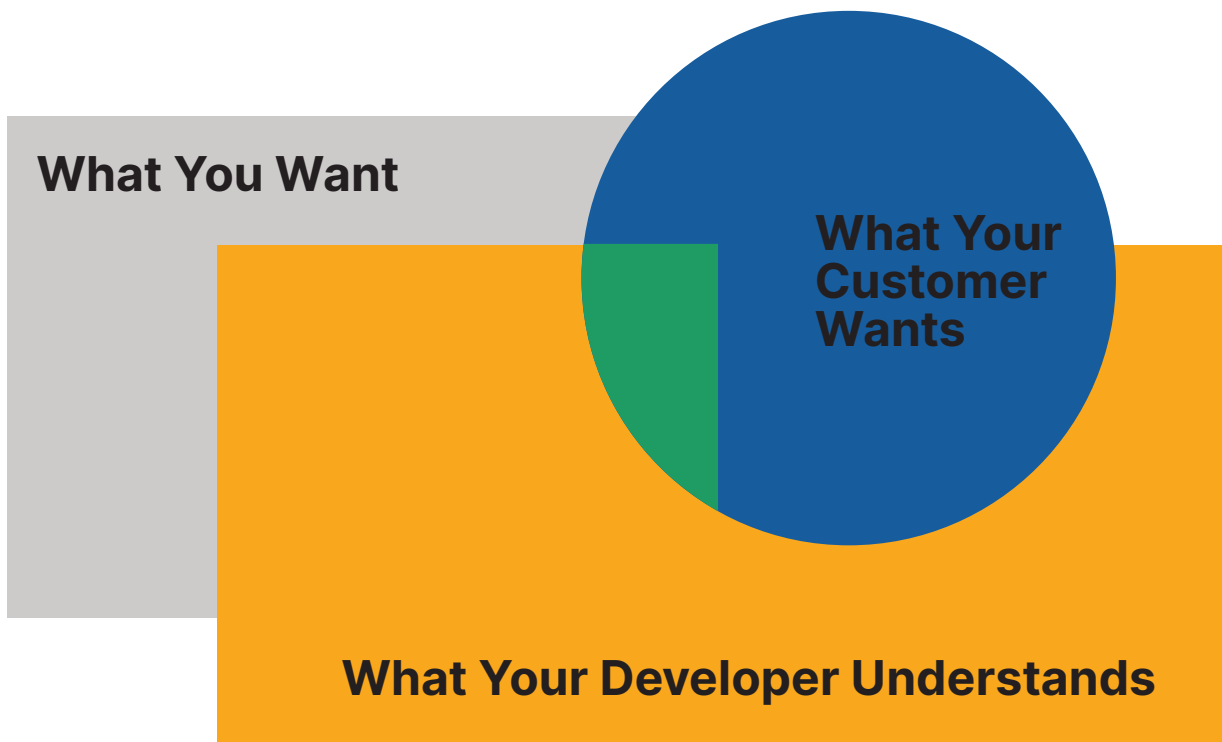
Biggest Mistake

Why Software Is Hard

#DevDifferent

Frame Misalignment

Owners, engineers, and customers have different mental models of the features that need to be build.



Edge Cases

Edge cases aren't considered during initial estimation, increasing the scope.

Unknown Unknowns

Hidden complexity with a task or technology.

Moving Target:

Requirements change mid-development due to new information learned. Expectations of what the completed item should look like change, but aren't communicated to the entire team. Forcing rewrites and bloated software.



How to Fix

User Stories, Usability Testing, rIDE, Become Technical

"Progress happens when it takes less time and resources to do the same task you did yesterday."

Technical Chasm

Why Software Is Hard

#DevDifferent

Technical Chasm

Most people are either non-technical, or very-technical, causing issues and opportunities in between.



Language Barrier

The larger the gap, the more issues there will be in communication.

Held Hostage

Non-technical founders feel like they are held hostage and often cycle through many different developers.



How to Fix

Become More Technical, Use AI to Communicate, Use rIDE to Plan and Task.

"Progress happens when it takes less time and resources to do the same task you did yesterday."

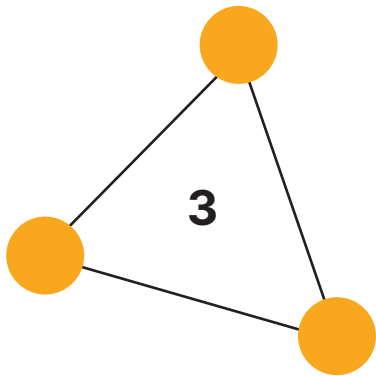
Technical Chasm

Why Software Is Hard

#DevDifferent

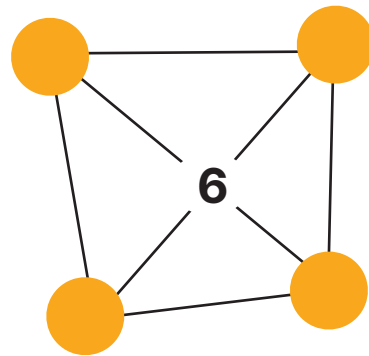
Mythical Man Month

Throwing additional manpower or forcing more meetings on developers does not speed up development; it slows it down. This is due to individuals stepping on each other's toes, breaking stuff that was once working, and requiring additional layers of communication.



Three Person Team

A three person team will have three lines of communication and there is less chance that they will be working on the same areas of code.



Four Person Team

A four person team will double the required lines of communication from three to six and make it much more likely that they will be working on the same lines of code.



How to Fix

Small Teams, Leverage AI, Leverage rIDE Code Editor and Experts, Micro-Contracting.

"Progress happens when it takes less time and resources to do the same task you did yesterday."

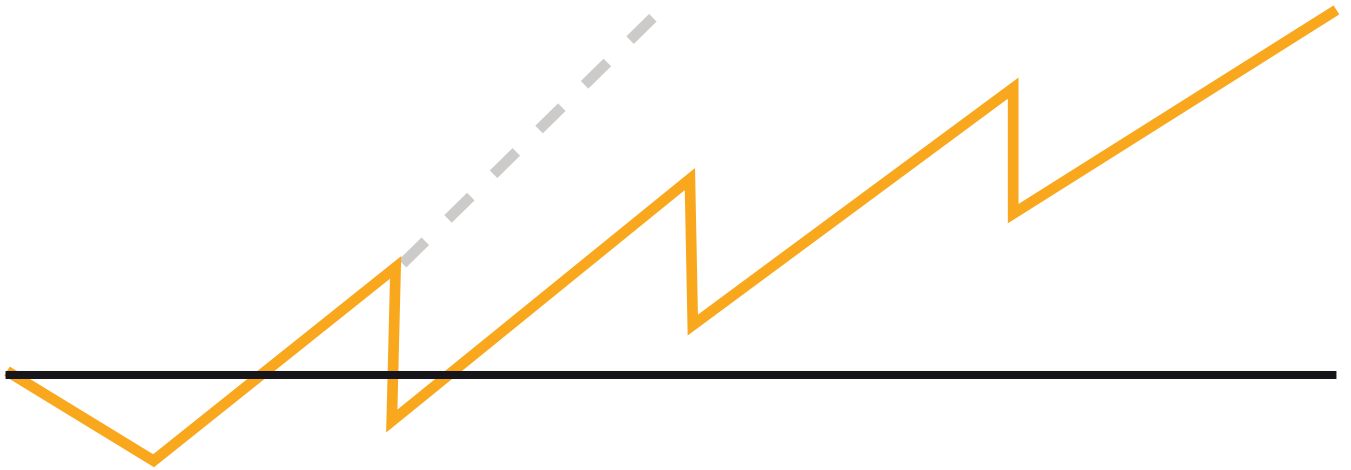
Technical Chasm

Why Software Is Hard

#DevDifferent

Brain Drain

New team members require training and increased communication overhead, slowing progress instead of accelerating it. When they leave the organization all that knowledge goes with them and you have to start again.



I Like It My Way

Every developer wants to build in their favorite tech stack, causing things to get unnecessarily rebuilt when teams change.

Brain Drain

Development teams churn, causing loss of information that was stored in a developer's mind and not on paper.



How To Fix

Documentation, Building System Context in rIDE, Psychological Safety, Usability Testing

"Progress happens when it takes less time and resources to do the same task you did yesterday."

Case Studies

#Dev Different

#DevDifferent

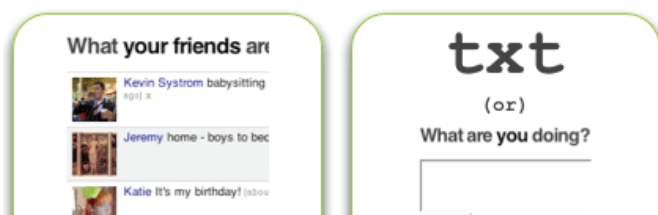
Ugly Starts

You don't want the polished version at first because polished versions cause you and your customers to have a psychological attachment. Customers won't want to tell you what they really think, and you won't want to replace things you've spent a lot of time perfecting.

Start with fewer requirements and focus on functionality over form first. Start with User Stories, Pencil & Paper, Wireframes. Avoid high-resolution mockups.



Use twtr to stay in touch with your friends all the time. If you have a cell and can txt, you'll never be bored again...E V E R !



Sign in.

Mobile number (or email)

Password (or PIN)

☐ Remember me

Sign in



Twitter (now X) was originally called "twtr" and all it began with is a text field where you could type in your phone number and what you are doing.

"Progress happens when it takes less time and resources to do the same task you did yesterday."

Technical Chasm

#Dev Different

#DevDifferent

Don't Design Too Far Ahead

Projects that are designed too far out rarely ever last the test of time due to lessons learned along the way. They end up wasting money and creating sunk costs that are psychologically impossible to get over.

Think only one or two sprints ahead.



"A feature deployed is worth more than 100 mockups."

Waiting On Design

Developers end up waiting long periods of time on designs.

Sunk Costs

Fancy designs make it less likely to get honest feedback and less likely for teams to want to make changes to the designs.

"Progress happens when it takes less time and resources to do the same task you did yesterday."

Product Lifecycle

#Dev Different

##DevDifferent

Retrospective Design

Traditional Design

1

High Resolution Mockups

Once you have a functional feature, you want to generate high resolution mockups, ideally using a program like Figma or Adobe XD.

2

Build Functional Feature

After you have user stories you can build out a functional version of the feature designed by a developer.

Retro Design

1

Build Functional Feature

After you have user stories you can build out a functional version of the feature designed by a developer.

2

High Resolution Mockups

Once you have a functional feature, you want to generate high resolution mockups, ideally using a program like Figma or Adobe XD.

3

Re-Design Completed Feature

Once you have high resolution mockups, you can go back and make your functional feature look better.

The modern rule of competition is that whoever learns fastest wins.

Wizard of Oz

#Dev Different

#DevDifferent

Wizard of Oz

Users interact with what appears to be a fully functioning system, but certain parts of the system are actually operated manually behind the scenes by a human.



Zappos Story

In the late 1990s, Zappos co-founder Nick Swinmurn had the idea of selling shoes online, but he wasn't sure if people would actually want to buy shoes over the internet.

Swinmurn went to local shoe stores, took pictures of shoes, and then posted those pictures on his website. When a customer placed an order, instead of shipping from a warehouse or inventory, he would go back to the store, buy the shoes at full price, and then ship them to the customer.

From the customer's perspective, it seemed like a seamless online shopping experience, but behind the scenes, it was manual and low-tech. The company lost a lot of money fulfilling orders but validated a billion dollar business.

"Progress happens when it takes less time and resources to do the same task you did yesterday."

Product Lifecycle

#Dev Different

#DevDifferent

Frame Alignment

Focus On Why Behind Each Feature

- 1 One of the biggest traps in software is to be too prescriptive. A sign of being too prescriptive is that whenever a problem is identified, folks are immediate to identify what a solution looks like and not take time to understand the why behind each problem.

Work in User Stories

- 2 One of the best ways to communicate functionality is a user story. As a [user] I want to be able to [take an action] so that [I receive some value].

Work in Short Iterative Feedback Cycles

- 3 The less time between completion and feedback, the more likely you are to catch issues of frame misalignment.

Identify Acceptance Criteria and Edge Cases

- 4 Make sure with each task to clearly define what the acceptance criteria are for that task (i.e. what success looks like) and also try to brainstorm what are the possible edge cases. An example of an edge case is what if someone uploads a video that is too large to process.

Use AI to Provide Details & Discuss

- 5 Utilize AI tools to help provide clarity and detail to tasks. Just make sure you review the results before sending them to your developers.

"Progress happens when it takes less time and resources to do the same task you did yesterday."

Technical Chasm

#Dev Different

#DevDifferent

Small Teams + AI + Experts

As teams grow, the number of communication paths and interdependencies increases exponentially, making coordination more difficult and inefficient.

Instead of having a large team of equally skilled developers, an optimal team consists of a primary lead (like a surgeon) supported by tools like RIDE.

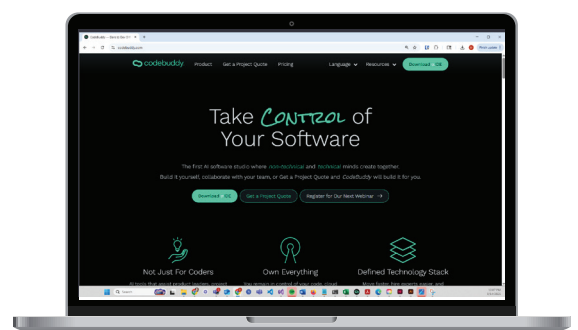


Artificial Intelligence

Utilizing artificial intelligence software tools like rIDE can help turn an a three person team into a team of 9.

Experts & Micro-Contracting

Utilizing experts and micro-contracting to fill in the missing technical pieces is a quick and easy way to accelerate your dev process and conquer difficult tasks.



The modern rule of competition is that whoever learns fastest wins.

Surveying the room

#Dev Different

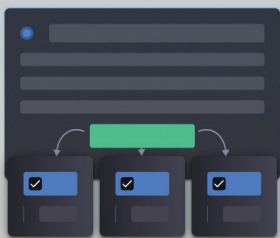
It's like going to a senior developer and saying
"I know you've mastered this type of task, can
you show me how."

Better Software Faster

Micro-Contracting

Using outside experts to show you how to do a **complex** task so your in-house system experts can build faster and smarter.

1

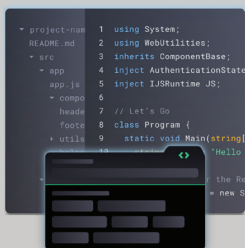


Breakdown Tasks

Breakdown your software tasks into areas that an outside expert could help with.

Example: I need a seasoned Azure cloud specialist to show me how to programmatically upload an image to Azure Blob Storage.

2



Find an Expert

Pay an expert developer to create a standalone prototype to show you how to do that task.

You don't need to share any of your code with them or spend time on-boarding them. They just need to understand your requirements and technology stack.

3



Build Faster

Your in-house system experts use the prototypes to complete the task in record time.

Your system experts who you trust to write quality code use the prototype as a learning tool to implement the task incredibly fast and with new expert insights.

All In-House
Developers
(Quality)

Micro-
Contracting
(Quality &
Speed)

Outsourced
Development
(Speed)

10 hours → 2 hours

Case studies show that with Micro-Contracting a 10 hour task can be done in under 2 hours.

"People ignore design that ignores people." - Frank Chimero

Surveying the room

#Dev Different

Building The Right Features

Usability Testing

Usability: Design that a person of below average skill in your ideal customer profile can use without it being more trouble than it's worth.



Usability Interviews: Watching people try to use what you created while they talk about it out loud.

Who to Test

You want to test people that are in your ideal customer profile and preferably don't know you, your company, and most importantly have never heard of your product.

How Many

You want to test 3-5 individuals in every round of usability testing.

"86% of user issues can be discovered with just 3 individuals."

- Forrester

How to Find Them

Ask your friends, family, and advisors if they know somebody. Post on forums, social media sites, and even freelancers sites. It's better if you don't have to pay them but either can work.

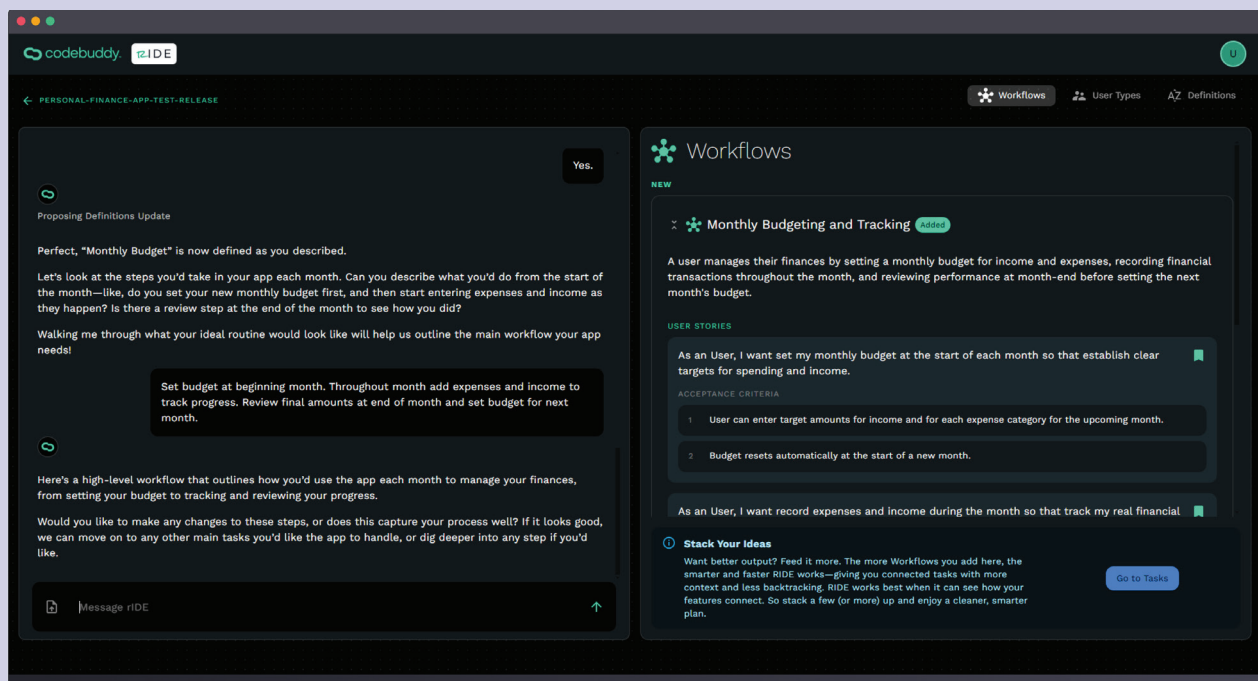


The First AI Software Studio Where Less-Technical and Technical Minds Create Together.

Build it yourself, collaborate with your team, or Get a Project Quote and CodeBuddy will build it for you. Breaking barriers to building new software.

Software Planning

Plan your software using AI specifically designed to breakdown and store everything about your application.



Build Faster

Plan your tasks and strategy in a fraction of the time.



Spend Less

Avoid the need to hire business analysts.



Expert Insights

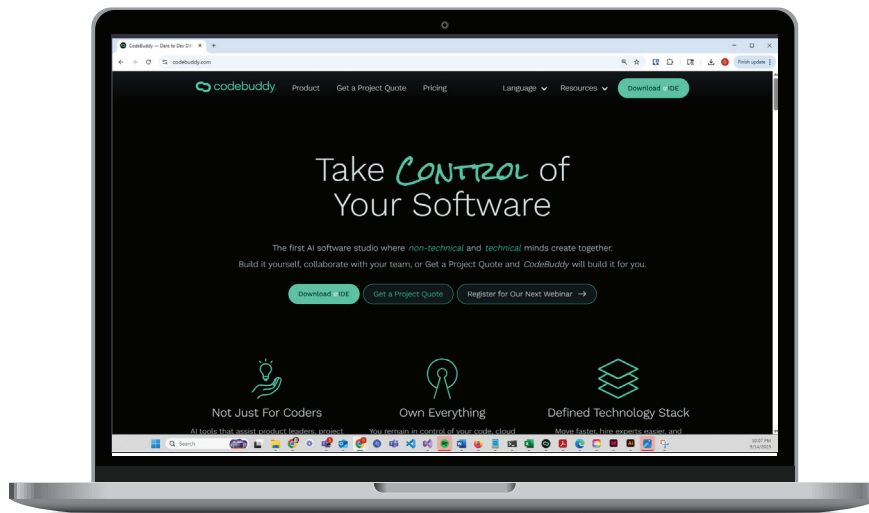
Provide greater detail to your development team.



Communication

Prevent frame misalignment and errors in scope.

Explore CodeBuddy

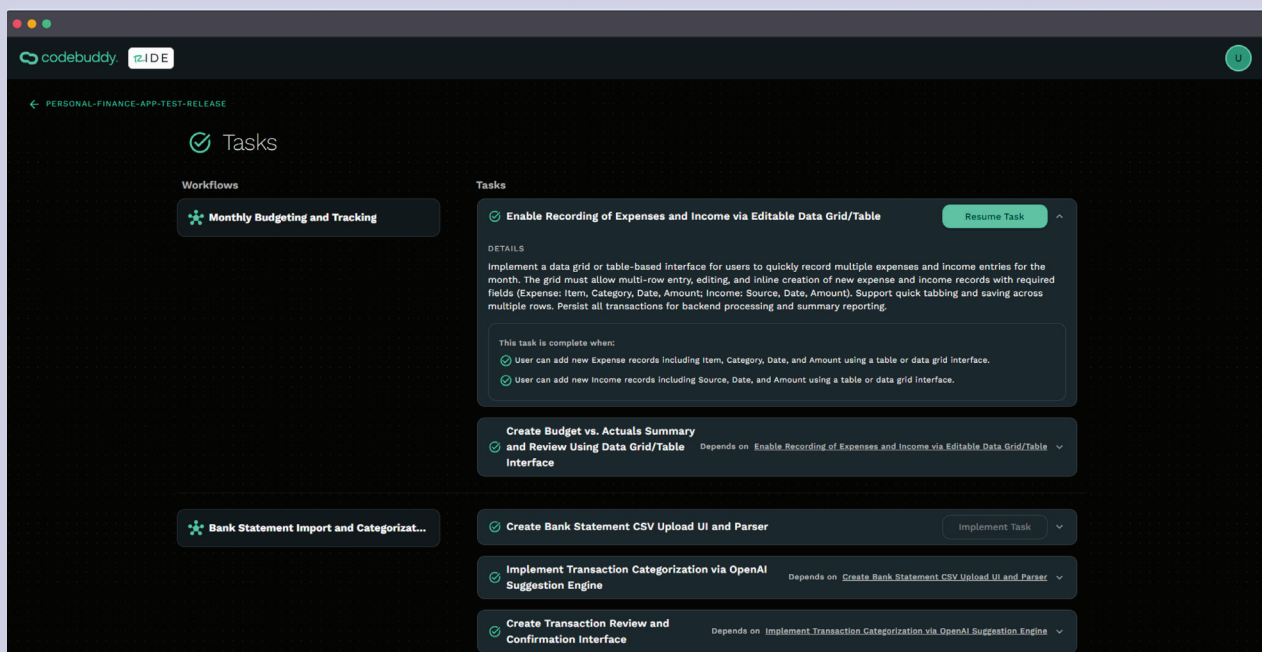


Take Control

The one software platform designed to put you in the driver seat of your software journey regardless of your technical expertise.

Task Planning

AI driven task planning and project management helps communicate the technical requirements for any given software task.



Build Faster

Manage your tasks in a fraction of the time.



Spend Less

Avoid the need to hire costly project managers.



Expert Insights

Provide greater technical details to your developers.



Retain Developers

Support your developers so they don't burn out.

HOW WE DELIVER ON OUR PROMISE



STARTERKIT TEMPLATE

Months of “on-par” software features and code already ready to go.



ONE TECH STACK

We don't try to do everything. We are really good at one powerful tech stack.



ARTIFICIAL INTELLIGENCE

AI is faster than any coder and we've been working on chat dev tools prior to ChatGPT.



CODEBUDDY EXPERTS

On demand professionals trained in our systems that can jump in and help.

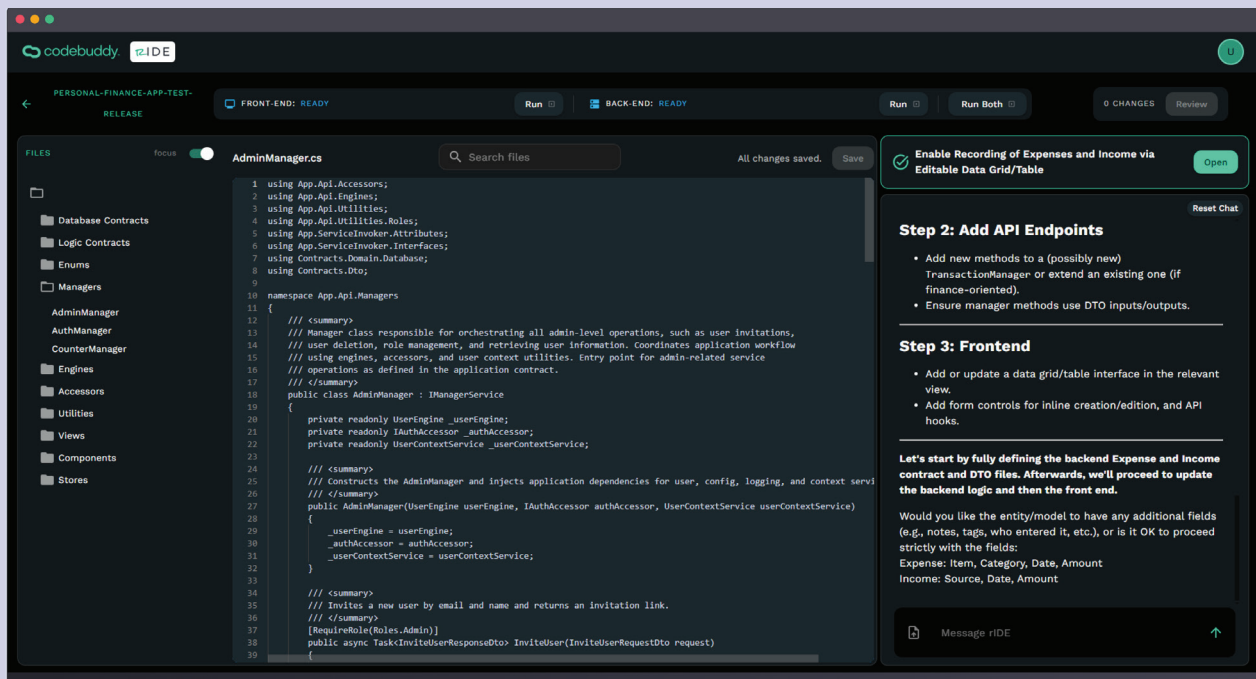


OWNER CONTROLLED

Maintain full access to all your code, cloud resources, data, passwords, and permissions.

Code Editing

Build your software with the guided help of our AI Code Editing tool. No other tools necessary.



Build Faster

Spend time reviewing code instead of writing it.



Spend Less

Build a high quality app with fewer developers.



Expert Insights

Improve code quality with expert insights.



Retain Developers

Support your developers so they don't burn out.

ZERO TO LAUNCH

CODEBUDDY TAKES ANYONE FROM ZERO CODING KNOWLEDGE AND TOOLS TO TO DEPLOYING THEIR FIRST SOFTWARE APP



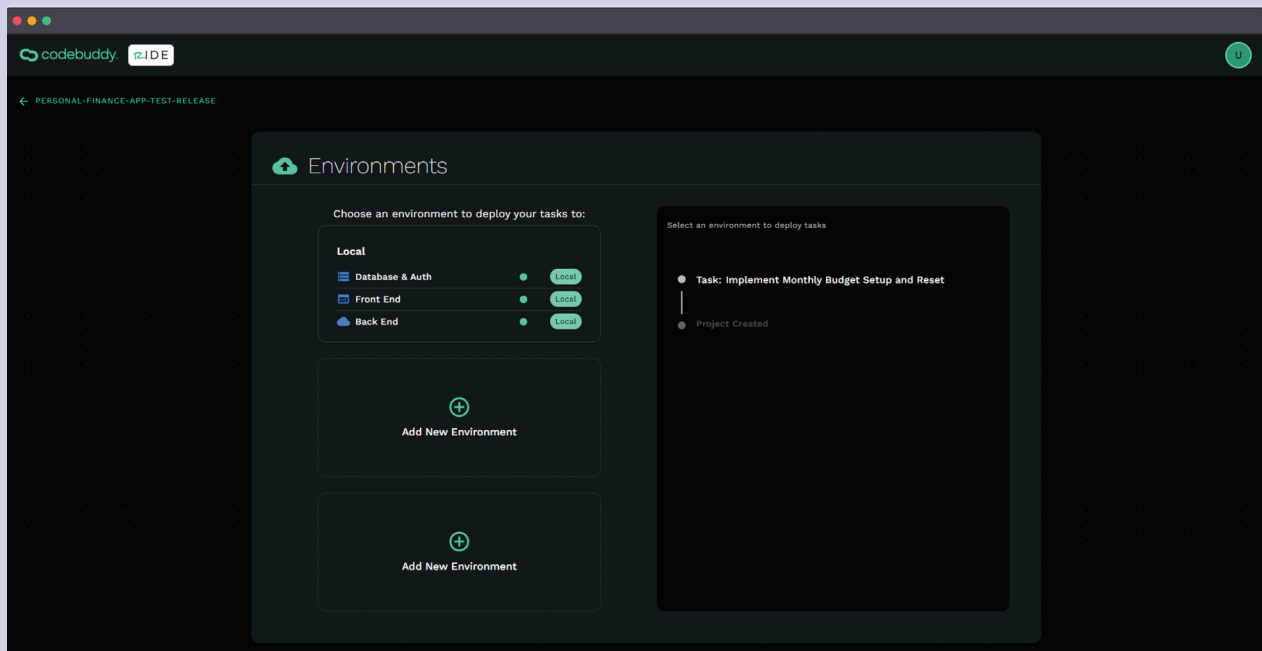
COMPETITORS

Get help from expert CodeBuddies anywhere along the way.



Publish Software

Publish your software to multiple test and production environments accessible to your employees or customers.



Build Faster

Deploy your site within minutes of building it.



Spend Less

Avoid having to hire costly dev ops people.



Expert Insights

Multiple environments allow you to test your app before you ship



Retain Developers

Support your developers so they don't burn out.



CodeBuddy helps developers complete complex coding tasks
5x faster through the power of crowd-sourcing and AI.

Devon Seacrest, CEO
devon@codebuddy.com



codebuddy.com