Designer's Guide to Developer Collab—oration

InVision

Twilio SendGrid

Table of Contents

This guide is brought to you by InVision and Twilio SendGrid.

InVision gives companies everything they need to power a collaborative workflow between designers and developers. 100% of the Fortune 100–brands like IKEA, Slack and Netflix– use our platform to build products customers love. Interested in a demo? **Speak with InVision today.**

Twilio SendGrid helps you focus on your business without the cost and complexity of owning and maintaining an email infrastructure. And with a full-featured marketing email service that offers a flexible workflow, powerful list segmentation, and actionable analytics, all of your email needs are met in one simple platform. **Start sending with us today.**

Introduction

The world's top brands have embraced design as a critical component of corporate strategy. However, **design without development won't get you very far**. To build design-rich content—from emails to websites to landing pages and everything in between—you need to collaborate effectively with the developers in your life.

While most designers and developers know success is dependent on a collaborative relationship, bridging the divide isn't straightforward. Roles, responsibilities, work style, and processes can contribute to miscommunication between these two parties—but it doesn't have to be this way.

With empathy, creative technology, and a few best practices, designers can foster harmonious developer relationships. With a spirit of collaboration and the right processes, these teams can implement designs that fuel better customer experiences and produce solid business results more than ever before.

In this guide we'll uncover best practices and deceptively simple ways you can boost collaboration with the developers you work with. From rethinking the designer-developer "handoff" to recalibrating your design pace—we'll arm you with everything you need to know to make your developer partner your new BFF in engineering.

Why all the hullabaloo about designer-developer relations?

Relationships matter. It's that simple.

"When everyone is aligned on the product vision and scope, it's much easier to keep a project on track," says Jennie Baird, VP of Product at News Corp. "Successful teams come together. They have chemistry. They have a shared vision that everyone believes in and is working towards."

As a designer, you can't do it all. Well, with some basic coding knowledge and a little know-how, you can—but you won't be nearly as effective as with a cross-team effort.

"When you know someone doesn't respect you, there is no trust," says Natalya Shelburne, Senior Software Engineer Technical Lead at The New York Times. "And when there's no trust or autonomy, people don't do creative work."

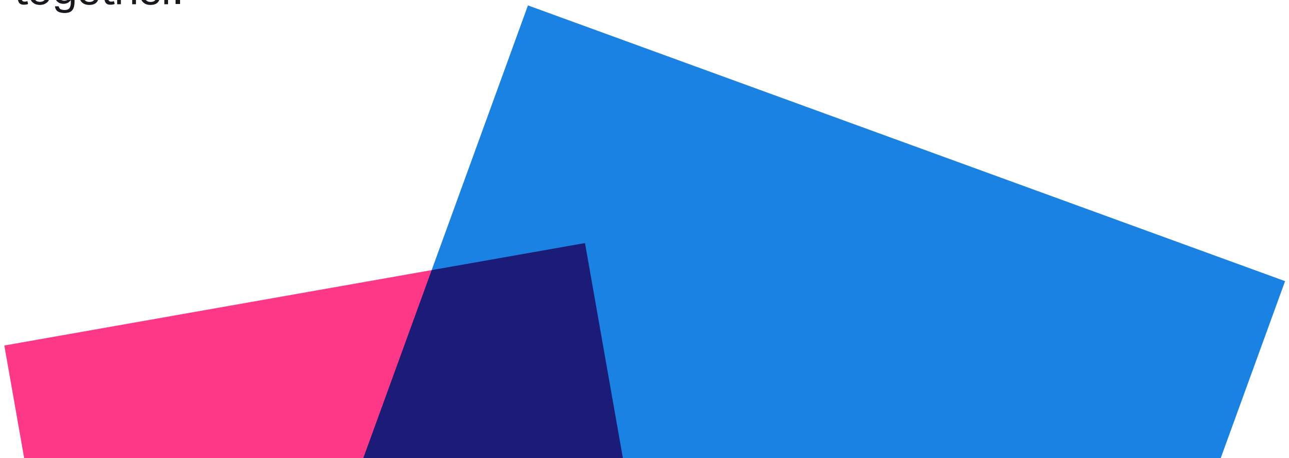
"One of the most important things for designers is to have a strong collaborative partnership with engineers. No great design work happens without that partnership being in place," says Alissa Briggs, Director of Design, Construction at Autodesk.

These relationships aren't just built around kind words and active listening, though—they're forged through thoughtful action. According to *Twilio SendGrid's State of the Marketer-Developer Relationship in 2019*, friction is created when marketers and designers fail to appreciate the scope of work required for a given request or bring the developers into planning discussions far too late in the process. Designers face these same challenges when working with developers.

Actions speak louder than words.

To deliver top-notch creative work, designers and developers need relationships centered on trust, appreciation, and respect. "There's a lot that can happen in building rapport with engineers," says Andy Law, former Director of Design at Netflix and now Head of Design and Research at Zoom. "You build empathy out of spending time with people and understanding their perspectives... so when we're working through a project, we have those relationships to rely on."

Designers and developers have to stop thinking "us" and "them" and start thinking "we." Because once these teams learn to come together and collaborate effectively as one, the sky's the limit to what they can design, prototype, and build together.



The optimal collaboration workflow

Every company and every team will work a bit differently, but **there are a few essential elements you'll need to create an optimal connected workflow between design and engineering.** Your processes may be casual or codified—regardless, these workflow steps will help.

1. Start with priorities

It's easy to get into "that's not my job" arguments when there's no context for a specific project. Start with the big picture. What's the ultimate goal of this email template, landing page, user interface, or mobile app? Is it to improve the user experience, boost functionality, rebrand, or something else?

When kicking off a new project, consider using Atlassian's Roles and Responsibilities play. This play helps to define everyone's duties from the get-go so you have crystal-clear expectations throughout the project.

When teams gain context, they start to see beyond their respective work. Designers begin to see that it might be worth scaling back the amazing visuals to improve page load time to ultimately improve the experience for the end-user. And developers may realize that it's worth the extra effort to create more responsive designs to accommodate different devices and clients.

Start with priorities, and many of your conflicts will sort themselves out naturally.

2. Create (and maintain) a cohesive team

When it comes to design and implementation, some designers and developers are going to butt heads—that's why it's important to build relationships outside of projects. Whenever possible, create harmonious teams with members who'll work well together—not just a hodgepodge of talented designers and developers.

When working on projects, you don't always get to choose the team you work with. Sometimes you have to play the hand you've been dealt. **When bringing a new team together, consider the 4 stages of team formation:**

Forming: Team members are generally polite and positive, but there's still some anxious energy in the air as roles are defined and relationships are formed.

Storming: As time goes on, conflicts eventually arise, and these are dealt with in the storming process. Work styles are questioned and stress enters the equation.

Norming: Team members resolve their differences and begin to find where they each fit in the group puzzle. Often, teams will cycle between storming and norming from time to time as they resolve emerging conflicts and establish new normals.

Performing: Performing is the stage where processes, structures, and relationships work without friction. Teams thrive at this stage, delivering their best work with the least amount of stress.

Once you've formed your dream team, keep the cohesive vibes going. Meet regularly and keep all parties in the loop. Maintaining a close-knit team with frequent communication will prevent technical impracticalities from road-blocking any projects.

3. Request development and provide all the specs

Bring your development team into the process as early as possible—don't wait until you've finalized your designs to get feedback from your friends in engineering. Initiate the Hot Potato process from the get-go (more on that in a bit).

It's better to ask for everything upfront than ask for this, that, and more down the road. When you approach your developers, provide all the specs you'll need. Don't make the engineers dig through emails, Slack conversations, documentation, and more to find the information they need.

InVision Inspect makes it easy to turn your designs into action for developers. It translates all your design files into detailed specs, so your developers can find measurements, colors, font styles, and code snippets. Transform how you collaborate with your best-buds in engineering with Inspect—**learn more here.**

4. Use the right tools for the job

InVision allows you to collaborate, experiment, and test efficiency and effectively. Whether you're using Freehand, the real-time digital whiteboard, to brainstorm with the team, Prototype to transform your static designs into clickable prototypes, or Design System Manager (DSM) to connect your design and code, you need a synergetic platform that make the workload easier for you and your developer pals.

Twilio SendGrid's Marketing Campaigns, for example, lets you build emails with visual drag & drop WYSIWYG editors or feature-rich HTML code editing. Plus, once your developers build the responsive templates, it's easy for you to edit pixel by pixel without breaking structural code.

5. Meet, ideate, and improve for next time

The workflow shouldn't end with the project. To improve future collaborations, you'll need to have a post-mortem of sorts. You can learn more about post-mortems in the free book Principles of Product Design.

Meet together as a team and hash out (politely) the good, bad, and ugly:

- What went well?
- What went wrong?
- How can we prevent this hiccup from happening in the future?
- Why did this roadblock occur? How can we avoid it?
- Did we meet our deadline? Why or why not?
- What are our learnings and takeaways?

The purpose of this meeting is not to point the finger of shame and blame. It's to **discover gaps and friction so that you can collectively plan how to patch them up and prevent future issues.**

Post-mortems and retrospectives (retros) aren't the same thing. Post-mortems happen after a project is completed, while retrospectives happen during the process. Retrospectives encourage change to occur throughout the process to improve the final result.

Now, you're ready to tackle the next project. Hopefully, each successful (and unsuccessful) collaboration will increase harmony between your designers and developers and ultimately produce the optimal workflow.

Effective Designer-Developer Collaboration Makes a Big Impact

Seems like a lot of work. Is it worth it?

We think so, and here's why:

Boost in performance: Better collaborations lead to better products. And better products lead to happier customers. What's more important than that? Research shows that businesses with highly engaged employees outperform their competitors by 147%.

Accelerated process: Teams that work together get things done quicker and better. Using design thinking to align design and development teams can accelerate workflows by as much as 2x.

Improved work culture: When you transition from "them" to "us," your work culture will improve for the better. A positive team environment builds trust, security, and collaboration—all essential elements of effective design. Google's Project Aristotle found that psychological safety (trust) is one of the most important factors in highly effective teams.

Creative solutions: When designers and developers work together, hybrid solutions are created that don't just get the job done—they evolve the way things are done forever.

No, it's not easy to build topnotch designer-developer relations. Yet, the things that aren't easy are usually the things worth fighting for.

However, there's no *perfect* model for designers and developers to work together. No two teams are the same, and no two businesses are identical. The collaborative environment you foster and build might not look like another business's successful partnership, but they'll likely share a few characteristics.

Whether you're in a startup with a single designer and developer or a complex corporation with a spider web hierarchy, a few universal best practices will improve your collaboration.

8 Best practices for forging better collaboration with your developers

1. Involve developers early in the process

Involve your developers as soon as possible—preferably, from the very beginning. Even if you're just in the brainstorming process, their input can be invaluable in terms of direction, solution, and scheduling. Don't wait until it's too late to bring them into the discussion.

“Share early and share often,” says Shafi Cassim, formerly at Atlassian and now Senior Product Designer at Chegg Inc. “A good workflow is to bring them [developers] in quite early, even at the artifact stages—to bring in the team so they get an understanding of the mental model, the framework, that I’m trying to get at with the designs.”

2. Step outside of your silo

"It’s not, 'Well, this is my part, and that’s your part.' And then the two shall never meet. It’s let’s put everybody everywhere," says Dan Mall, founder and director of SuperFriendly. Just because you're a designer doesn't mean you can't contribute to developer conversations. And just because your developer has a degree in computer science doesn't mean they can't offer helpful design advice.

Step outside of your silo and embrace the other side. Learn the fundamentals of code so you can be more specific with your requests and expectations. And when possible, empower developers to step outside of their silo and get a real taste of that sweet designer life.

3. Understand not everyone fits a mold

Yes, in a perfect world, you'd have a quintessential model to follow that's made up of demi-god designers and developers who can do no wrong. But that's not the world we live in.

In our world, you have brick people and mortar people. The brick people have clear boundaries—like a back-end developer who only touches the back end. However, it's tough to build a wall with only bricks on bricks.

You need mortar. And the mortar people are the multi-talented designers and developers (or design engineers—more on that in a minute) who help fill in all the cracks. They're malleable and not afraid to tinker, leave their comfort zones, and get their hands dirty with a touch of the unknown.

Not everyone has to do everything—some people just do one thing very well, but they don't dabble in other aspects. While it'd be nice if they could, that's just not always going to happen. That's the world we live in, and it's your responsibility to learn how to work with brick and mortar people to build a cohesive team.

To learn more about brick and mortar people and how they work together, check out the Brad Frost and Dan Mall: Rethinking designer-developer collaboration episode of the Design Better Podcast.

4. Play “hot potato" instead of executing a one-and-done handoff

In the game of Hot Potato, the object (rarely an actual potato) gets passed back and forth quickly—you a little, the next person a little, someone else a little, you a little again, and so on. This is how your handoff to developers should go: a little bit of design here, a little bit of development there, back to design, back to development, and back and forth until the project is done.

However, most teams treat the handoff as a one-and-done deal. You design, they develop—no going back and forth unless something goes epically wrong. This idea leads to heavy design, and then no design. Then it leads to heavy development, and then no development. One-and-done handoffs strive to increase efficiency by eliminating multiple touchpoints and ongoing communication, but this usually avoids any sort of collaboration.

Hot Potato, on the other hand, can be clunky and inefficient at first. You'll lose a bit of efficiency upfront, but you'll return much more in quality later on. **You don't have to go wholly one way or the other—just try to incorporate a bit more back and forth into your handoffs rather than a "Here you go, and I'm out of here" approach.**

Discover more about the Hot Potato theory in the Brad Frost and Dan Mall: Rethinking designer-developer collaboration episode on the Design Better Podcast.

5. Always prioritize the customer

Many designer-developer struggles revolve around preferences, opinions, respect, defined roles and responsibilities, and the like. However, these factors forget the end purpose of any project: the end-user experience.

In the end, it doesn't matter who's right, who comes up with the solution, or who gets the credit—it just matters that the end-user loves the final result.

Whenever there's a disagreement or roadblock, ask yourself if it's about the end-user or if it's more about the individuals involved in the design and implementation. If it's the individuals, bring the team back to focus on what really matters. If it's the end-user, then work out (preferably with data, when available) the best option to optimize the user experience.

Let the engineers learn firsthand what the customers want. Involve them in the user research so they can hear from the mouths of the end users. Margaret Lee, Director of UX at Google, set up a program to teach developers basic research techniques and to spend time talking with customers. It was through this program that Google was "able to, at scale, get the engineering org in Google Maps completely bought into thinking about the user first," says Margaret Lee.

6. Focus on MLPs, not MVPs

An MVP (or minimum viable product) is the basic version of your product or design with just enough features to satisfy beta testers and early customers without overcommitting resources. It's much easier to revise than it is to create, so designers have focused on getting MVPs out the door as soon as possible.

However, there's an even better way to get workable products out into the market. Introducing the MLP, or minimum loveable product. This approach focuses on getting an MVP out the door but with the design-rich (and feature essential) elements that'll get customers excited and talking.

"It's better to build something that a small number of users love, than a large number of users like," says Sam Altman of Y Combinator. These are words you should live by as you develop your MLPs.

7. Consider adding a design engineer to the team

The design engineer role isn't a one-size-fits-all solution for every team, but it does help effectively bridge the gap between designers and developers. A design engineer (also known as creative technologist, UI engineer, UX developer, etc.) is a hybrid player that can design and code.

When trying to build designer-developer collaborations, you can see why this role would be useful. The design engineer truly understands the pain points for both teams and can act as the ultimate liaison. However, as you'd expect, it's not all frosted donuts and pumpkin lattes.

Instead of using a design engineer to bridge gaps, some teams offload all the collaborative portions to this hybrid role. Anything that might be an overlap between design and development becomes the design engineer's responsibility, and then collaboration falls solely on their shoulders rather than the collective teams. You can learn more in the free book Design Engineering Handbook featuring insights from design leaders at *The New York Times*, Mailchimp, Minted, and Indeed.

8. Speak in layman's terms

Your developers haven't lived in the world of creative design like you. Speak in a language that they can understand.

Don't go throwing around terms like ascenders, kerning, orphans, and widows and expect a developer to follow along. Go down that rabbit hole, and your friends in engineering might just respond with thinking, exceptions, pointers, and Boolean expressions. Yeah, best to keep things simple and straightforward for everyone whenever possible.

However, you can impress your friends in engineering (and earn a dose of respect) by going the extra mile and learning how to speak in the language of developers. UX Designer at Google, Laura Martini, shares the value of speaking in the language of developers as a designer. She says, "As designers, we are really good at discovering our users needs, but don't always apply that same rigor within our own company." You can hear more from Laura in this interview. Do your research and learn the basics. Just as country residents respect foreigners who attempt to speak the local language, developers will appreciate you trying to understand their terms.

Next steps for making it happen

Now that you understand the importance of designer-developer collaboration and how work effectively, you're ready to start making it happen. Here's how:

1. Get on the same page: First, share this guide with your fellow developers and dedicate time to follow up and chat. After some back and forth and a few head nods, you'll hopefully all be on the same page moving forward.

2. Adopt tools that foster collaboration: Conversations alone can't always bridge the designer-developer divide—you need tools that not only enable collaboration but encourage and support it.

InVision DSM: Design System Manager connects design and code so teams can work smarter, faster, and more in sync. DSM seamlessly works with the tools designers already use. Integrations with Sketch Libraries lets you upload files in a single click, sync changes, and push and pull design system assets.

InVision Inspect: Inspect is where design and development work like magic. Turn design into code in a snap with powerful spec creation, in-design feedback, and developer workflow integrations. Get started absolutely free here.

Twilio SendGrid: Use simple drag-and-drop tools, pure HTML, or a mix of both for complete control of your email marketing campaigns. Build email designs and workflows that have limitless potential with scalable email APIs that developers love. Get started with a free plan here.

3. Try it out: Your collaboration efforts won't be perfect at first, and that's okay. Try, fail, learn, try again, succeed, learn, and keep trying. Improved designer-developer collaboration comes from intentional efforts over a period of time—don't give up!

Start improving your collaboration now

Designer-developer collaboration isn't easy, but it's worth it. As Natalya Shelburne writes in the Design Engineering Handbook: "The hard truth is that the effort to facilitate effective workflows between disciplines is as real and challenging as the rest of our work." There's no quick fix—you'll need to invest time, energy, and emotions into making this work. By using the tips and best practices in this guide, you can forever improve the way you work with developers.

Remember, there is no one-size-fits-all collaboration solution or perfect workflow. As Natalya says, it's about "shifting expectations, surfacing patterns, and building systems to facilitate collaboration."

Arm-in-arm with your fellow developers, there's nothing you can't tackle together.

About Twilio SendGrid

We send emails for 82,000 customers (using over 2 trillion email addresses!) and have been in the email game since 2009, so we understand that email is the perfect combination of people and tech. We know that email can be the key to developing a successful business model, and we support our clients as they create meaningful campaigns.

No matter your background in the industry or role in your business, we know that your email needs are unique, and a one-size-fits-all solution won't cut it. Twilio SendGrid provides customer communications solutions for every stage of your company's growth because, at the end of the day, it's not only how we communicate, but why we communicate.

Start sending with us today.

About InVision

InVision is the leading product design and development platform for teams building world-class digital products. Our platform and services enable remote collaboration across roles and time zones for improved speed to market and powerful business results.

More than 7 million people at both large enterprises and small startups come to InVision when they are looking for digital transformation. Our platform allows teams to ideate, prototype and test new ideas; create repeatable and streamlined processes in design, product and engineering; and up-level workflows to move more efficiently from inspiration to production.

100% of the Fortune 100–brands like IKEA, Slack and Netflix–use our platform to build products customers love. Interested in a demo?

Speak with InVision today.