A bookler book

Feed your read

1

The Design Sprint — Google Ventures

The Design Sprint

The sprint gives teams a shortcut to learning without building and launching.

The sprint is a five-day process for answering critical business questions through design, prototyping, and testing ideas with customers. Developed at Google Ventures, it's a "greatest hits" of business strategy, innovation, behavior science, design thinking, and more — packaged into a battle-tested process that any team can use.

Working together with companies in a sprint, we shortcut the usual endless-debate cycle and compress months of time into a single week. Instead of waiting to launch a minimal product to understand if an idea is any good, teams get great data from a prototype. The sprint gives these companies a superpower: The ability to build and test nearly any idea in just 40 hours.

This page is a DIY guide for running your own sprint. On Monday, you'll unpack the problem. On Tuesday, you'll sketch competing solutions on paper. On Wednesday, you'll argue and decide how to turn your ideas into a testable hypothesis. On Thursday, you'll hammer out a high-fidelity prototype. And on Friday, you'll test it with real live humans.

2

How to Run Live User Testing, Part 1: Setup — Medium

At Cluster, we're big fans of iteration and experimentation. Since we launched publicly in February 2013, we have rapidly iterated the product on both iOS and Android. In the first 8 weeks of being live in the Apple App Store, we launched 10 updates. On Android, there was a week that we pushed out 5 releases in 5 days while we did some heavy A/B testing.

While rapid iteration is wonderful, at times we also slow down and make more deliberate decisions about larger changes. When this happens, we tend to make rapid prototypes and then test them in front of different groups. Most of these tests are fairly informal, but occasionally (admittedly not often enough) we run full-blown user testing, where we recruit and bring in potential users to walk through the app and give us feedback.

We are working on a big update, so we recently ran multiple sessions for different prototypes. When talking about it with fellow entrepreneurs, they asked us for details. Here is our ever-evolving playbook.

Because this is fairly lengthy, we will write this in three parts:

- Setup
- Running the Tests
- Debriefing

Before getting started, I want to give a huge amount of thanks to Michael Margolis and the Google Ventures design team, who taught us most of these techniques.

Part 1: Setup

User tests are extremely valuable, and require a lot of work to get set up. All in all, it takes a significant amount of one person's time over the course of a week to setup the tests and run them well. So make time to do it correctly.

The setup process is divided into the following sections:

Decide on a specific thing to testDecide when and where to do the user studyDecide what type of users to studyRecruit users with CraigslistTrim the candidates listPrioritize and scheduleGet the right equipment

Let's get started!

Decide on a specific thing to test

We were recently lucky enough to do a sprint with the exceptionally talented design team from Google Ventures (entire post on that coming soon). Over the course of five days, we identified core opportunities with Cluster, brainstormed improvements, built several simple prototypes, and tested the prototypes with potential users. The result of this process was a clear idea of what new concepts were working and what wasn't.

With the list of successful ideas, the team then rapidly built a single functional prototype based off our current app. With this working prototype completed, it was time to show it to users and see if all the insights gathered from the design sprint worked in the context of our actual app.

Decide when and where to do the user study

I was scheduled to be spending a week in Nashville, which gave us a great opportunity to run our test outside of the Bay Area. This is incredibly valuable because the bay area tends to be filled with tech savvy, early adopters. Nashville has its fair share of them, but technology isn't as core to the community there, so we felt like this would give us an opportunity to meet more "real" users.

It's also important to pick a quiet, private, and neutral place. As tempting as it might be to use your company's conference room, I'd recommend not bringing users into the corporate office. Use a friend's office or co-working space. All advice we've been given is the more neutral the location, the better.

The privacy and quietness is important because you'll be recording the session, so you don't want to do it in a coffee shop where there's a lot of distraction and background noise.

Nashville Entrepreneurial Center

When I was in Nashville, I rented a conference room from the Entrepreneur Center. It worked out perfectly.

Decide what type of users to study

This is a very very important part of the process. Before recruiting users, you need to decide what type of people you want to meet with. We had done this during the Google sprint, and because our app involves sharing photos, we asked these types of questions:

- What type of phone should they have?- How involved with social media should they be?- What apps should they use (and not use)?- How many photos should they take per week?- How old are they?- How do they share photos currently?

With these questions in mind, we created a Google Forms survey that would help us clearly identify if the potential tester fit our target profile. It's not worth user testing if you're not testing the right type of user, so take some time and do this step properly.

The actual screener we used for our Nashville study

Since the user is going to have to physically be somewhere, it's also useful to get their availability. We did that by starting with the question saying "Which of the following times are you available on Thursday February 6 to come to downtown Nashville?" with 5 options for time slots.

Recruit users with Craigslist

About a week before we planned to do the user testing, we posted a job opportunity to the jobs/et-cetera section of Craigslist. In this relatively short post, we give very little information, except we are looking for people to participate in a usability study, they'll need to be okay signing an NDA and being filmed, and we are willing to pay them for their time (in this case, a $75 Amazon gift card for a 60 minute meeting).

Craigslist post driving people to the screener

The post did not let users email reply. Instead, there was a link to a Google form we built above. That made it super easy for consolidate and organize everyone's responses.

Trim the candidates list

Usually, we'll get between 60-200 applicants within a few days. We try to pare that down to 5. This happens over several rounds of editing.

For this study, the first big cuts happened with device type. Because of the small amount of time we had to build the prototype, we only could test with iPhone users that had iOS 7 installed, and ideally an iPhone 5 and up because we didn't have time to optimize for all screen sizes. Although this skewed the users a bit, we were able to re-balance it by looking at the other info.

Example responses from the Nashville screener, with all personal information removed

We then eliminated anyone who didn't take photos with their phone. Although it might be interesting to talk to these users eventually, we were looking for people who would have an immediate reason to use our app. If they didn't take photos, it was unlikely they'd be the type of user we wanted anyway.

Prioritize and schedule

With the remaining candidates, we looked at their age, occupation, and a couple other data points and put together a prioritized list of the people we were most interested in talking to. At this point it became a scheduling exercise, slotting our top pick for each time slot and choosing a backup if that person couldn't make it.

Each candidate was emailed saying they'd been selected for a time slot and they needed to write back within a certain time frame to confirm, or their slot would be given away. The backup list was emailed saying they were on the backup list and to let us know if they no longer could make it if picked, otherwise they would hear by a certain time if they were needed.

Moving candidates around while scheduling

As a warning, people are VERY flakey. Out of the 5 top candidates, only 3 confirmed, and one of them cancelled the day of. It wasn't a problem because we were able to fill in the slots with our backups, but it's a bigger pain that you'd expect. It's even wise to have multiple backups just in case.

Get the right equipment and software

We were testing a mobile app, so it was important to record the user actually using the app. Although you can do this by plugging the app in and watching a screencast on the computer, it's much better to actually see them touching their phone. We purchased a $100 camera for this, and it's well worth the investment.

IPEVO Ziggi-HD High-Definition USB Document Camera

The only other thing you'll need is a way to record the audio and video of the session to your computer. For this, I recommend an excellent app called Screenflow.

3

The GV research sprint: Interview participants and summarize findings (day 4) | Google Ventures

At Google Ventures, we have a four-day process for answering questions and testing assumptions without the time or expense of launching. We call it a GV research sprint. This is the final in a series of five articles on running your own research sprint. (You can also watch a 90-minute video about research sprints.)

Research sprint checklist:

- Create a recruiting screener
- Post recruiting screener where the right people will see it
- Select and schedule participants
- Start creating interview guide
- Confirm participants
- Complete interview guide
- Review prototype with your team
- Set up test devices and recording system
- Interview five customers!
- Summarize findings and plan next steps with your team

It's showtime! Today you get to talk to real people — your research participants — and test your prototypes. Your interviews will help you answer big questions, test your assumptions, and figure out what to do next.

By the start of Day 4, you should have five participants scheduled to come in, an interview guide, and one or more prototypes to test. Now it's time to set up your room and your equipment, prepare yourself for the interviews, conduct the actual interviews, and summarize with your team.

Set up the room

Before participants arrive, use this checklist to prepare:

- Print copies of the non-disclosure agreement (in case participants did not e-sign).
- Print your interview guide (for your reference).
- Clean up the interview room: erase whiteboards, throw out empty coffee cups, etc.
- Post "do not disturb" sign on the door of the room.
- Gather incentives for participants (gift cards, etc).
- Grab a notepad and pens for any short notes you want to take. (But your team will handle the bulk of the notetaking.)

- Make sure there's a clock on the wall or table.
- Put a box of tissues in the room, just in case.

Set up your test devices and recording system

It's best to keep your research technology as simple (and cheap!) as possible — that'll give you a lot of flexibility and minimize technical difficulties. You'll probably want two different versions: one for testing desktop prototypes and one for mobile prototypes.

Here's how I do it.

Testing desktop prototypes

For testing desktop prototypes, I use a MacBook Pro with GoToMeeting. If you are testing prototypes of websites or web apps, Mac or Windows will work fine — just test with a web browser that your customers would realistically use. But if you are testing prototypes of desktop apps or operating systems, you'll need a computer and OS that matches the people, scenarios, and platforms you are designing for.

I always use an external keyboard, mouse, and display with the MacBook, in case my participants are not familiar with the MacBook trackpad or keyboard. Make sure the display is a realistic size — it's common for designers and engineers to use 27-inch or larger displays, but this might not be true for your target users.

I put a small webcam on top of the display to capture video of the participant's face. I also connect a second mouse to the MacBook so I can easily take control to point something out or get the participant unstuck (when appropriate).

I use GoToMeeting for streaming and recording the screen and video of the participant's face. (It can also be used for remote interviews.) I've tried many products over the years — including Google Hangouts, Apple AirPlay, and WebEx — and I believe GoToMeeting has the best combination of reliability, video/audio quality, and features. During the interviews, my team can join the "meeting" and watch from another room.

After setting up your test computer, use this checklist to prepare:

- Test video streaming and recording.
- If using GoToMeeting, schedule a "meeting" for each interview. Add the meeting URLs and phone numbers to your team's calendar.
- Install the prototypes and do a test run. (Be sure to revert prototypes to their default state before the interviews.)
- Hide obvious personal information and remove desktop icons from computer.
- Clear web browser history, cookies, cache, and bookmarks.
- Create bookmarks or shortcuts for prototypes.
- Set an innocuous home page (e.g. your local newspaper).
- Minimize the browser or prototypes until you're ready for your participants to test them.
- Wipe down the keyboard, mouse, and display.

Testing mobile prototypes

For testing mobile prototypes, I still use a MacBook Pro and GoToMeeting, but I add the Ziggi document camera to capture video of the iPhone, iPad, or other

test device we're using. Set the mobile device on a table and position the document camera above it. I still use a webcam for video of the participant's face, but the document camera is doing the heavy lifting here. The Ziggi works with a simple app called IPEVO Presenter, which displays the video on my screen. When this app is open, I can simply share my screen via GoToMeeting and my team can see the video feed as well as the webcam.

(For a similar setup from our portfolio company Pocket, see "How we built a research lab for mobile app testing in just a few hours.")

After setting up your computer and mobile test device, use this checklist to prepare:

- Test video streaming and recording.
- If using GoToMeeting, schedule a "meeting" for each interview. Add the meeting URLs and phone numbers to your team's calendar.
- Install the prototypes and do a test run. (Be sure to revert prototypes to their default state before the interviews.)
- Clear web browser history, cookies, cache, and bookmarks.
- Create bookmarks for prototypes or save to home screen.
- Close the browser or prototypes until you're ready for your participants to test them.
- Open the IPEVO Presenter app on the computer.
- Position the Ziggi document camera so it's focused on the test device.
- Position the webcam so it's focused on the participant's face.

- Wipe down the keyboard, mouse, and display.

Prepare your team to watch the interviews

While you're interviewing participants in one room, your team — ideally the entire team who worked on the prototypes — will be in another room observing the interviews and taking notes.

Put one of your team members in charge of the observation room. Make sure they read these two articles before Day 4 begins:

The GV Design Sprint: Validate (Day 5) — from our series on the Google Ventures design sprint

How to Run Live User Testing, Part 3: The Debrief — from our portfolio company Cluster

This guide is not meant for the observers (although those other two links are), but there are a handful of important reminders you'll want to give your team:

- Grab plenty of snacks and drinks so they don't have to step out during the interviews.
- Test the audio and video before the interviews begin.
- Mute their microphone (and video camera, if necessary) so participants don't get nervous about being watched.
- Assign one note-taker to write detailed, transcript-style notes — and rotate so no one is stuck typing for five hours!
- Stock up on Post-It notes, notepads, pens, and whiteboard markers.
- Pay attention and write down observations, insights, and ideas during the interviews.
- Keep laptops closed and phones put away during the interviews.
- Collect insights and ideas after each interview, so it doesn't all pile up at the end of the day.
- Have fun and keep an open mind!

Get into character

With your equipment set up and your team ready to go, take a minute to prepare yourself.

Help your participants feel comfortable by skipping perfume or cologne, avoiding onions or garlic at breakfast, and chewing mints or gum before each interview. Dress the way your participants would dress. (For example, don't wear a T-shirt when interviewing attorneys or physicians. Don't wear a tie when interviewing truck drivers.)

Minimize distractions by silencing your phone and turn off any notifications on your computer. Take a bathroom break before each interview.

Finally, get into character by adopting a curious, open, and objective attitude. Take a deep breath. Smile. Try to see the world through your participants' eyes. And resist the temptation to judge or dismiss what they say — remember, your goal is to elicit and learn from your participants' real reactions to your questions and prototypes.

Showtime!

As you begin your interviews, here are a some tips to keep in mind:

- Be a good host. Smile. (Yes, I know I'm repeating myself, but smiles and mints will get you far!)
- Build rapport. Make small talk and ask easy-to-answer background questions. Don't just jump right to the prototypes.
- Let participants figure it out on their own. Don't pitch or explain the product. Answer questions with questions.
- Ask who, what, where, when, why, and how questions.
- Ask follow-up questions. Silence and pregnant pauses are also effective ways to draw people out.
- Avoid leading questions.
- Ask for specific, recent, personal examples — instead of theories or projections about other people.
- Mostly be quiet. Let participants do the talking.
- Watch the time.
- Watch for participants' non-verbal cues. Are they nervous? Frustrated? Annoyed? Try to hack your body language for better interviews.
- Go overboard in expressing appreciation for their time and input. "Thank you sooo much! This has been incredibly interesting and valuable."

For more advice on conducting great interviews, check out these 16 tips.

Between interviews (especially after the first one), check with your team to see if they have any questions about what just happened, or suggestions for the next interview.

Take a break before your next interview. Conducting five interviews in a day can be exhausting.

Then reset the prototypes and the room, get into character, and greet your next participant.

Summarize, review, and plan your next steps

After the last interview, don't let your team leave before meeting to summarize what you learned and plan the next steps. If they took proper notes, this

shouldn't take any more than 30 minutes — which is good, because you're probably exhausted by now!

Your main job as a team is to highlight the patterns (both good and bad), consolidate and organize those patterns, and collect ideas for possible solutions to any problems you found. Brenden Mulligan from GV portfolio company Cluster does a great job of showing how to do this. (Don't forget to capture the whiteboards by taking photos.)

With your team (and your research findings!) in one place, figure out what you're going to do next. Usually, you'll want to update the prototype to fix some problems, create a new higher-fidelity prototype, or decide to focus on a new set of questions and assumptions to tackle. You'll almost certainly want to plan another design sprint to continue designing your product.

Then — finally — you can call it a day. Scheduling your next research sprint can wait until tomorrow.

Thanks for reading all the way to the end. We've helped more than 100 startups use research sprints to learn more, make better decisions, and move faster. We can't wait to hear about your sprints! Let us know how it goes — tweet us at @GVDesignTeam or @mmargolis.

GV research sprint series

Previous: Finalize schedule and complete interview guide (day 3) Next: That's it, you're done!

4

Got A Bright Idea? Test It With A Rapid-Fire User Study | Co.Design | business + design

[Editor's note: This is the sixth in a series of seven posts on running your own Google Ventures design sprint. Missed the other previous posts? Here are Parts I, II, III, IV, V, and VI.]

At the Google Ventures Design Studio, we have a five-day process for taking a product or feature from design through prototyping and testing. We call it a product design sprint.

As day 5 of the product design sprint dawns and the team files into the war room, there's a certain something in the air. Is it the chemical scent of the white-board markers? Maybe. Is it coffee breath? Yeah, almost for sure—in fact, you might want to hand out some gum.

But there's also something else: anticipation. Today, we test our prototype and learn which ideas worked, which didn't, and what to do next.

Today, you're going to be running a user study, showing your prototype to four to six real humans. By "real humans," I mean people who don't work at your company—more specifically, people who you'd like to have as users. Learn how to recruit great participants for your study here.

Wait, what if I don't know how to run a user study?

Bad news: This post won't teach you how to talk to users. This post is for the people watching the study, not for the poor soul conducting the interviews.

Good news: If you're the interviewer, you can get some great advice from our research guide. Be sure to check out Michael Margolis' articles on interviewing tips and body-language hacks, and his video about running user studies from the Google Ventures Startup Lab.

And you might as well read this post, too, so you know what's going on in that observation room. I went to all the trouble of typing the whole thing up, so it's kind of the least you could do.

Start at the end. When the day is over, you'll want a concrete list of ways to improve your prototype. Talking about your game plan before the study starts will help you get there.

The interviewer and the observers should make a list of the key questions for the day. Here are a few tips:

• Review your conflicts and assumptions. Those assumptions you flagged as testable with a user study? Now's the time!

• Are you testing multiple prototypes in a battle royale? If so, be sure the interviewer understands the differences between the two versions, so they can ask the right questions.

• Consider showing participants some real products for comparison—they're like free prototypes!

• What else do you want to see through your users' eyes? Today's study is not a usability test —you have a chance to find out how your users understand your product, what competitors or substitutes they use, and more. Think beyond the prototype and you are guaranteed to get some unexpected insights.

Everybody who participated in the sprint should be in the room. There's no substitute for watching real humans use your product, and this is a golden opportunity to do it!

I like to reserve two rooms for the day: one for the user interviews, and another where the sprint team can watch live video of the sessions and take notes together. It's kind of like a Battlestar Galactica marathon, only instead of comparing guesses on who's a Cylon, you'll be comparing guesses on which parts of your design are going to go up in flames. It's actually pretty fun.

Test the A/V ahead of time Setting up live video for observation shouldn't be too hard—at the Google Ventures Design Studio, we've had success with WebEx, GoToMeeting, Apple Airplay, and Google Hangouts.

Just be sure to test your video before you start the study—ideally the night before and the morning of—because nothing's worse than putting in all this work and not getting to watch the outcome. OK, there are a few worse things. But you see what I mean.

A few more A/V tips: • A conferencing mic can provide a huge improvement in sound quality. • Don't forget to mute the audio in the observation room! • Quickly re-test the A/V between each session. Seriously, stuff gets wacky.

Don't diss the user If you see people struggle to understand the prototype, keep in mind that we're testing your design—not the participant. If they don't get it, it's not because they're dumb. It's because you haven't nailed the design yet.

Make it clear in the observation room that it's not OK to diss the participant. It's tough to wade through a prototype while people are watching. You owe the participants your respect.

Every observer takes notes Everybody should take notes on things they see during the interviews: good, bad, and other. Insist on paper note-taking—it's best to keep laptops closed, lest you lose your fellow observers to email.

Designate a court reporter One person can use a laptop in the observation room, but they have a tough job: the court reporter. They're responsible for typing a word-for-word transcript of the interview in real time. Assign a different court reporter for each session—it's cruel and unusual to make the same person do it all day.

It's kind of a pain, but these notes become an incredibly valuable reference after the study, and a text document is a heck of a lot faster to scan for quotes and reactions than a video recording. Often times, we don't record the study and rely on these notes instead.

Make a scoreboard Clear one big whiteboard to collect the group's notes. Make a column for each participant and a row for each part of the interview (e.g., background, first prototype, second prototype, etc.).

At the end of each session, write down the highlights from everybody's notes—you can double check any questions against the transcript. I like to color code: green for things that went well, red for problems, and black for everything else. It makes it easier to find patterns at the end of the day.

I've never been in mission control when one of those rover things lands on Mars (NASA isn't returning my calls). But I imagine it's pretty similar to the atmosphere in the observation room of a user study. There's tension and excitement and nobody wants to be the one who designed the doohickey that blows up. Here's what you might expect to feel throughout the day.

First session: "We're geniuses!" or "We're idiots!" The first user may love your prototype. Or they may hate it. Either way, take a deep breath. People are different, and I offer you an ironclad guarantee that not everyone will react in exactly the same way.

Regardless of how that first interview goes, resist the urge to make changes to your prototype. Unless it's something simple like a typo or broken link, you risk "fixing" something that the next four users would have actually liked.

Sessions 2–4: "Oh, this is complicated…" It's not uncommon for the second and third participants to have dramatically different feedback, which means you're going to feel a little confused. Just sit tight and keep taking notes. It's still too soon to tell.

Actually, that's not entirely true. Sometimes you know for sure that part of your prototype is rotten after two or three interviews, and watching more people suffer through it is punishment for all involved. If everyone agrees that something is way off, talk to the interviewer in between sessions and ask them to guide people around that part of the prototype.

Studies 5-6: "There's a pattern!" After the final interviews it's easy to see the big patterns, but it's worth double checking the notes. Go to the scoreboard and look for things that showed up two or more times. Mark good stuff with a big green dot, and bad stuff with red.

Now make two lists on the whiteboard: "things that work" and "problems to solve." These are your top-line findings. The CEO or decider for the project should bless that list before you leave the room.

The sprint is complete. Take a deep breath.

The vast majority of these studies ends with mixed results. Some of your solutions work, and some don't. The outcome usually falls in one of three buckets:

A. Most stuff worked This is pretty uncommon for the first sprint on a project, but if it happens to you, everyone on the team is probably on the same page about the fixes and tweaks you need to make.

What to do next: Tune your existing prototype and keep going. Try starting your next sprint at step 3, when you decided which prototypes to make.

B. Some big questions The most common outcome after a user study is a mixed bag: a few hits, a few tweaks, and a couple of real head-scratchers. Fortunately Keynote prototypes are easy to change, and as long as some parts of your design are solid, you can probably build on what you have.

What to do next: You can move fast on the tweaks, but you'll want to come up with multiple solutions for the bigger problems. Start your next sprint at step 2, creating a storyboard.

C. Everything exploded I've seen a lot of of my designs go up in flames. It's OK. You learned that something didn't work, and it only took you a few hours to build it in Keynote. This is great progress and—relative to building and launching for real—very cheap progress at that. Think what would have happened if you'd spent weeks or months implementing this solution!

What to do next: Start your next sprint back at the drawing board with step 1 (developing a team understanding). (Hint: the results of this study are perfect fodder for reviewing and building understanding as a group.)

You may feel tired. Actually, if you don't feel tired, you probably didn't do the sprint properly. It's an intense week. But you've built up a tremendous head

of steam—even if everything exploded, you now have a much better understanding of the problem and possible solutions.

At the end of a sprint, CEOs often tell us they appreciate that they have a clear list of what to do next. Now that you've built up all that knowledge and momentum—not to mention a prototype—you'll find you can do the next sprint more quickly, or with less effort, as long as you do it right away. Don't let more than another workday go by before you jump back into it. If this problem is important, you've got to bear down and finish it off before people get distracted.

- How To Conduct Your Own Google Ventures Design Sprint
- From Google Ventures, The 6 Ingredients You Need To Run A Design Sprint
- The First Step In A Design Challenge: Build Team Understanding
- The 8 Steps To Creating A Great Storyboard
- How To Decide What Ideas To Prototype
- A Lightning-Fast Way To Make A Digital Prototype

5

Story-centered design: how to make a prototype in PowerPoint | Google Ventures

When I started my career as a management consultant, I was surprised at how versatile PowerPoint was. We used it for everything from seating charts to market-entry strategies for Fortune 500 clients.

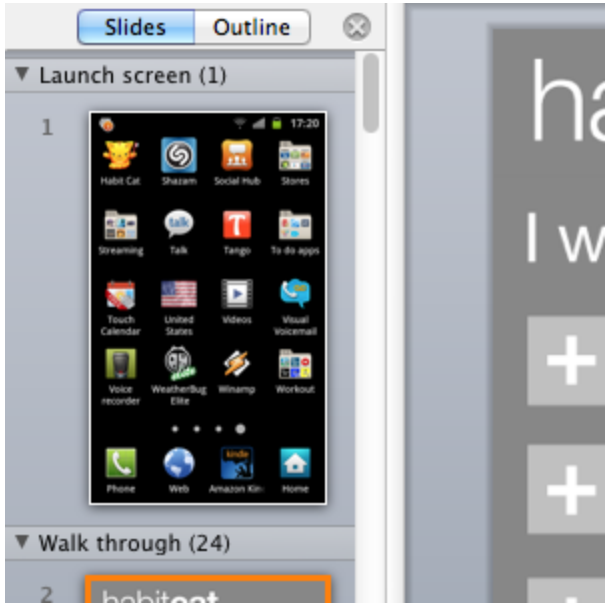
At Astrid, I found a new use for PowerPoint: building prototypes. These clickable mockups are great for rapidly testing with users and applying what you learn right away. They simulate the experience of using a website or app without having to write any code. Braden Kowitz described how the Google Ventures design team uses clickable mockups in a previous Design Staff article.

Here's an example of a prototype I created using the process below.

1. Prepare your slides

Start by creating a PowerPoint file and setting the appropriate size in page setup. The actual size doesn't matter as much as the aspect ratio — you'll want to match the aspect ratio of the platform you're designing for (desktop, tablet, or phone).

Create one slide for each state of your prototype. For example, a product setting with on and off states requires two slides. You'll end up with quite a few slides, so it's helpful to organize by adding sections in the thumbnail view on the left side of your screen.



2. Create the UI

Now it's time to create the UI. The most straightforward approach is to create your UI directly in PowerPoint — use templates like Keynotopia's or just build it from scratch.

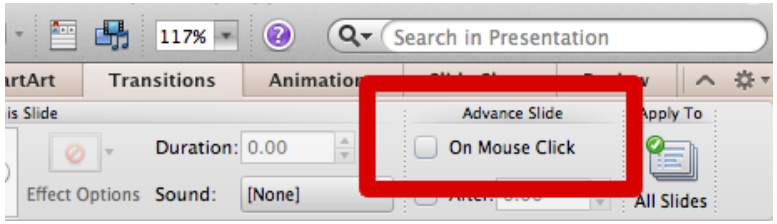
If you're really handy in Photoshop, Illustrator, or another app, you can create UI mockups there and make it clickable in PowerPoint. Here are some tips:

- Export common UI elements (like backgrounds, buttons, etc) separately and use them to build the necessary states in PowerPoint.
- Leave text out of your mockups and instead use PowerPoint's text boxes — this will help you quickly create different states and make changes after usability tests.
- Use transparent PNGs to simulate effects like Lightbox.
- To save time, you can export mockups as complete screens (instead of separate pieces), but it'll be much harder to make changes later.

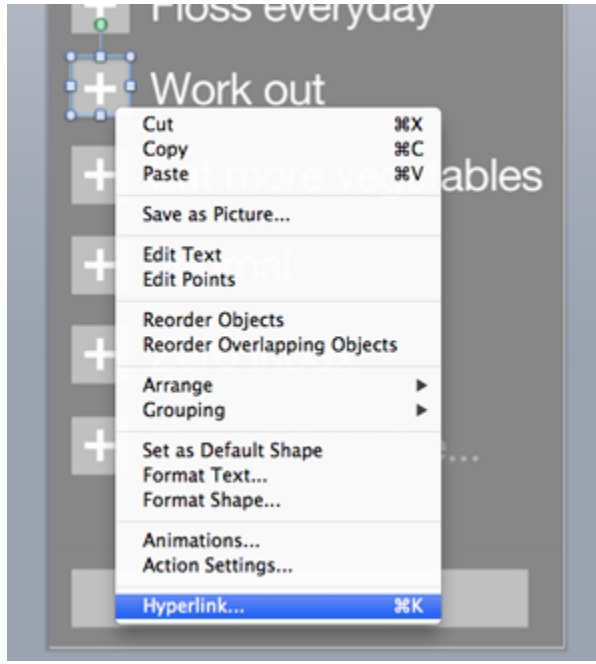
3. Make it interactive

Now the fun part: making your mockups come alive.

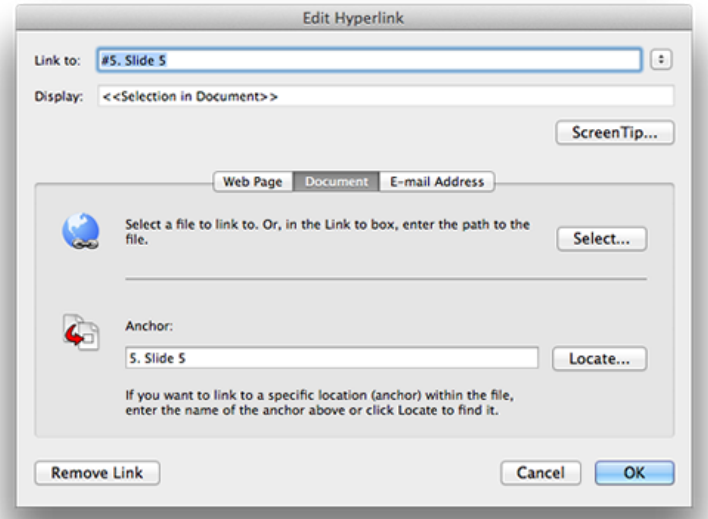
First, select all slides in the thumbnails pane and disable "Advance slide on mouse click" (under "Transitions"). Web pages and apps don't advance when you click just anywhere, so you'll want to emulate this behavior in your prototype.



Next, link clickable elements like buttons and menus to their appropriate destination slides by right clicking and selecting "Hyperlink." For example, if you have on and off states for a button, have the button link to the slide that shows the opposite state.



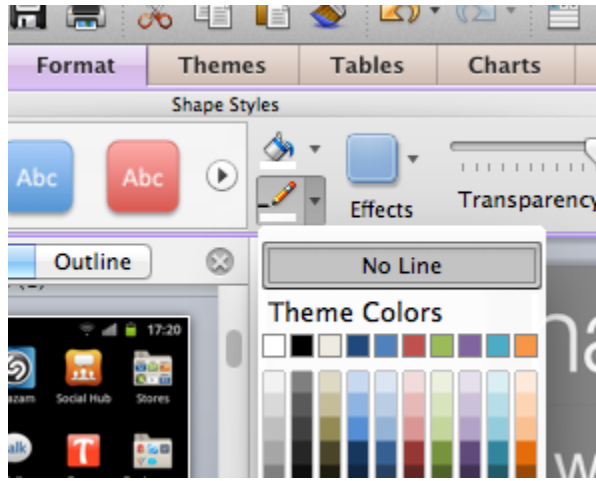
Link to specific slide numbers (for example, "Slide 5" instead of "Next slide") and the links will automatically update even if you move slides around or insert new slides.



4. Use these advanced techniques to make your prototype really shine

Invisible boxes

These inconspicuous heroes are great if you need to add a clickable area where there is no separate element to select. Use the rectangle shape tool to draw a box, double click, select no fill and no line, then link away.



Invisible boxes are also useful when a tappable area in an app needs to be bigger than the actual UI element. (For reference, Apple suggests tappable areas should be at least 44 points by 44 points.)

If you copy invisible boxes and paste them in another slide, the boxes will appear in the exact same position on the new slide. This can be handy when working on a series of slides that are variations of the same state.

Decoy screens

Certain situations call for special transitions. For example, a welcome walk-through for an app might have screens that swipe left as the user taps a "next" button.

In these cases, you can't simply apply animation to a slide — transitions are tied to the introduction of the second slide (rather than the exit of the first), so you'll trigger the same animation every time the slide loads. That'll mess up your hyperlinks and ruin the illusion.

The solution is to create a decoy slide. When you need special transitions, create two identical slides — the first is a decoy with the sole job of displaying the animation. Set the decoy slide to automatically advance to the second, identical slide. Then hyperlink to the second slide to avoid the transition when needed.

Learn before you launch with a quick study

When your prototype is finished, conducting user research can be as simple as firing up presentation mode and watching someone use it.

You can even use a prototype like this to simulate interactivity on a mobile phone. The Google Ventures design team uses Keynote to build prototypes, then exports a hyperlinked PDF and opens it in GoodReader on an iPhone. Research participants can click through the prototype like it's the real thing!

To learn more about user research, check out the GV guide to research.

Have you ever tested with clickable mockups? I'd love to hear about your experiences in the comments.

6

A Lightning-Fast Way To Make A Digital Prototype | Co.Design | business + design

[Editor's note: This is the sixth in a series of seven posts on running your own Google Ventures design sprint. Missed the other previous posts? Here are Parts I, II, III, IV, and V.]

At the Google Ventures Design Studio, we have a five-day process for taking a product or feature from design through prototyping and testing. We call it a product design sprint. This is the sixth in a series of seven posts on running your own design sprint.

On day 2 you drew concept sketches. On day 3, you made a plan and a storyboard for your prototype. Now it's day 4, and the clock is ticking. You're going to create a real-looking version of yesterday's storyboard and show it to users tomorrow.

This part of the sprint is super exciting for me as a designer. Thanks to the storyboard, I know exactly what to do, and I also have a crazy deadline to get it done. It's finally time to to open my laptop, put on my headphones, and start moving pixels.

But wait a second…what should this prototype look like? Are we going to have to code anything?

Quite simply, a prototype is anything a person can look at and respond to. A prototype doesn't usually have to be very complex in order to learn what you need to know.

Make it minimally real You'll probably be amazed at how much real feedback a user can give you on a slide deck of mockups that aren't even pixel-perfect.

They can tell you what they understand about your product—and what they don't. They'll tell you what they expect things to do ("I'd click here because I'd want to see a list of your customers…"), and when they get confused.

You'll also learn things that metrics alone can't tell you, in particular why users do the things they do, rather than just what they do. And you'll learn much faster than if you had waited to build something.

Write real text While it can be tempting to use "lorem ipsum" when you're building a quick prototype, don't do it—always write real text for your prototype.

When you use dummy text, you avoid tough decisions.

Why? First, user interfaces are mostly text. Punting on the text might save you time, but it won't get you closer to solving your problem. Plus, in design sprints we're often figuring out how to explain things and testing whether users understand. Lorem ipsum skips all of that. When you use dummy text, you avoid tough decisions and limit how much you can learn. (37signals has a great essay on using real words in their book Getting Real.)

Keynote versus code Occasionally you'll need to write some code for your prototype. This can be really valuable when you're testing assumptions relating to data quality or personal content like email. But nine times out of ten we find that Keynote is sufficient. More on this later.

Here are some examples We have created dozens of minimally real prototypes for the startups in our portfolio. Here are a few examples, covering the spectrum from grayscale and low-fidelity to nearly pixel-perfect.

If you're building a product for desktop, mobile, or iPad, the fastest and easiest prototyping tool you can use is Keynote. I've spent my career working with fancy tools like Photoshop and Illustrator and Fireworks. (OK, Fireworks might not be fancy, but it's almost certainly a tool.)

Here's why Keynote wins for prototyping:

• It's fast. • It's easy to make things look pretty good… • But it's impossible to make things look perfect, so you don't get too precious. • Anybody can quickly learn to use it; not just designers. • The slideshow format is a natural fit for story-based design. • It costs only $20. • The animated transitions are a lightning-fast way to make your prototype look way more real than it really is. • If it's a mobile prototype, you can export a PDF and open it on your device with reasonable results. When you're done, you not only have a prototype, you have a presentation, like, for free! • After your user study, when you learn all of the problems in your design, it'll take minutes to make changes instead of hours or days.

("Wait," you say, "I use a PC." Wow, you just blew my mind. Almost everybody I meet who works at a startup has a Mac, but true, there are exceptions. If I stereotyped you, I am deeply sorry. Let me make it up to you by pointing you to Henry Tsai's excellent post on building PowerPoint prototypes.)

The final Keynote (or PowerPoint) prototype might be linear or it might have a few links that allow the user to click around. It might be a grayscale wireframe or it might be somewhat polished. Either way, these make for good prototypes.

They're going to be realistic enough that your user study participants will forget they aren't clicking on a real product.

Keynotopia templates make it even faster Here's our favorite secret prototyping weapon: Keynotopia. It's a template with buttons, menus, and all kinds of plug-and-play elements that you can just drop into your prototype without any design skills at all. This makes it easier for CEOs and marketing directors to dive in and make something to communicate what they're thinking. And it's available for PowerPoint too, if that's how you roll.

All right already, I bought Keynote, now what?

You made a wise choice, my friend, and the prototypes are going to start flowing like milk and honey.

Here's a rough schedule for your day of prototyping. Expect to spend about an hour in the morning to review and make your plan.

Divide and conquer What if you have too much to design and not enough time? Work together, baby.

Take a look at your team. How many people can help in Keynote? Chances are good that almost everyone can—in our sprints, we frequently have engineers, PMs, and CEOs cranking out good work in Keynote. Remember that everybody's work doesn't have to be genius level. A designer can always clean it up afterwards, but it's usually faster to clean up a slide after someone else has put the building blocks in place.

Now take a look at yesterday's storyboard—or, if you're doing a "battle royale," storyboards plural. Break it into chunks and assign them out. It's probably obvious, but it's helpful to think about where different people have expertise, or where it's best to spend your designer's time, or who moves the fastest.

Reduce the impulse to check your email and burn up valuable make time.

You probably want to divvy up the best of the storyboard sketches on paper, too. Those are often the blueprint for many of the mockups you'll have to make, and they can save you a bunch of time.

Assign one person to be the stitcher. The stitcher's job is to take everybody else's work and put it into one cohesive flow. If you only have one designer, or if one person is really fast at Keynote, she might be a good candidate. If you're doing a "battle royale," you may decide to have one stitcher for each competing version.

Breaking up the design work and stitching it back together probably doesn't sound like the way great design is done. And it's not. Realistically you'll get better results if you've focused enough that your designers can do all the work

here. But if you need it, dividing and conquering is an incredibly fast way to build a good-enough-to-learn-from prototype.

Build an asset library Another way to increase efficiency is to take a few moments at the beginning of the day to build a template slide deck. Include anything that everyone will need—screenshots, user avatars, logos, formatted text; whatever you think might help. And don't forget to include a browser bar at the top for realism. You don't want to go through 99 slides at the end, scooting each one down to make room.

Use a timer to maintain focus Once again, the Time Timer is your friend—although if you don't have one, I guess some other kind of clock might work. We'll often say something like: "Let's work heads-down for the next hour and then we'll take a break." Time-boxing can make the project feel less daunting, and reduce the impulse to check your email and burn up valuable make time.

Appoint an email sheriff It's sometimes helpful to appoint someone—usually the facilitator—as the "email sheriff." This person's job is to publicly shame anyone who is checking their email (or surfing the web or whatever time-wasting tactic you prefer). It's usually enough to just make a big deal about saying that someone is going to do this…the threat alone practically makes the job unnecessary.

Lightning critique If you have a bunch of people working in parallel, it might be helpful to do a quick critique midday. It's useful to ensure consistency and get outside eyes on your design. But critiques can eat up a lot of time if you aren't careful. Just like on day 2, you're going to want to limit the time spent talking about each design. I recommend 5 minutes.

You're also going to want to prevent people from trying to redesign somebody else's work—they might have a great idea, but there just isn't time in this part of the sprint. Instead, use the lightning critique as an opportunity to raise questions and criticisms, but make it clear that the person who designed those components is going to be responsible for figuring out the solution, not the group.

Review with an outsider Schedule 30 minutes with someone who is not doing design work today. It might be your user researcher if you're lucky enough to have one, or it might be an executive—seems like everybody's got one or two of those hanging around. The outside eyes will help prevent you from going too far down any groupthink rabbit holes. Just make sure to do this early enough in the day that you have plenty of time to respond to the feedback afterward!

Outside eyes will help prevent you from going too far down any groupthink rabbit holes.

Pointers, text, and other final touches If you're creating a linear prototype with no links, which is the very fastest thing to do, it's a good idea to draw a mouse pointer and add extra slides showing where the user would click, and extra slides when text is entered. It takes only a little extra time but makes the user study much more realistic.

Other crucial details: Check for consistency and typos, especially if you've got some made up user data. Don't let it start out as Sally and end up as Suzy. That stuff is distracting in a user study.

Try to make any content current and relevant. If you're running the study in Seattle, and you need to show a newspaper, show the Seattle Times, not the Milwaukee Journal Sentinel.

If you get stuck, remember what you're trying to learn—don't waste 30 minutes tweaking a button style if you're doing a study about whether people understand the value proposition.

Nine times out of ten, you can learn everything you need to know in a user study with a click-through of mockups.

But sometimes you can't avoid it. When you're testing big assumptions involving data quality, or when users need to interact with their own stuff (email, docs, contacts, etc) to really understand the product, there's no substitute for a code prototype. In this case, get someone with engineering chops to help.

Just make sure everybody, especially the engineer, knows that this code will be throwaway. That's really important: The. Code. Is. THROWAWAY. We gotta move fast, so don't get attached. Like the Keynote prototype, it just has to look sort of real, it doesn't have to be anywhere near perfect.

It's better to be done with something good enough than to be half-finished with a masterpiece. Remember that the goal is to learn from the user study tomorrow, not to have everything perfectly figured out and finished.

Stay tuned for the next—and final!—installment of the series.

And in the meantime, what tricks do you use for super rapid prototyping? Do you have a tool that you swear is better than Keynote? I'd love to argue with you about it (respectfully, of course) in the comments!

- How To Conduct Your Own Google Ventures Design Sprint

- From Google Ventures, The 6 Ingredients You Need To Run A Design Sprint

- The First Step In A Design Challenge: Build Team Understanding

- The 8 Steps To Creating A Great Storyboard
- How To Decide What Ideas To Prototype

[Image: Digital via Shutterstock, Email via Shutterstock]

7

The GV research sprint: Schedule participants and draft interview guide (day 2) | Google Ventures

At Google Ventures, we have a four-day process for answering questions and testing assumptions without the time or expense of launching. We call it a GV research sprint. This is the third in a series of five articles on running your own research sprint. (You can also watch a 90-minute video about research sprints.)

Research sprint checklist:

- Create a recruiting screener
- Post recruiting screener where the right people will see it
- Select and schedule participants
- Start creating interview guide
- Confirm participants
- Complete interview guide
- Review prototype with your team
- Set up test devices and recording system
- Interview five customers!
- Summarize findings and plan next steps with your team

Select and schedule participants

After 24 hours, you should have a good set of participants to choose from in your Google spreadsheet. Review each one, and using your recruiting criteria from Day 1, highlight the ones that meet your criteria. When you encounter a response that's way off the mark, you can hide it or make it gray so you won't waste time looking at it later.

(If you used Google Forms, responses will be in a bona fide spreadsheet, so you can use filters, sorts, conditional formatting, and other features to automate this process. You can even throw some charts and graphs in there, for fun.)

Once you've narrowed down the list of potential participants, select the five candidates who fit your criteria the best. Then call them! Yes, on the phone. With a brief phone call, you can verify their screener responses, check that they're sufficiently articulate, confirm their availability, and start establishing rapport.

After the phone call, follow up with email to confirm the time, date, and location of the interview. Include:

- Any instructions that will help people arrive on time, such as directions, parking info, the nearest transit stop, and what to do when they arrive.
- Your phone number in case they have questions or need to reschedule.
- A link to a non-disclosure agreement that people can review and sign ahead of time. You can use DocuSign or HelloSign for online signatures, or else just attach a PDF. (In addition to confidentiality, this sample NDA gets permission to record and protects ownership of ideas).

To minimize frustrating no-shows, ask people to reply to confirm. Your email subject line could say: "Reply to Confirm — Usability session scheduled on May 10 at 1 pm."

This might seem like a lot of back-and-forth just to schedule participants, but every response will boost your confidence that they'll actually show up for the interview. You're moving fast, and only doing five interviews, so you'll want to make the most of your time.

Read more about how to minimize no-shows in the GV Library.

Start drafting interview guide

Now you can start planning what exactly you're going to do during the interviews. You could wing it, but you'll get much better results if you follow an interview guide. Why? With an interview guide you can:

- Think about how to phrase questions without "leading the witness."
- Create a checklist for important details and topics during the interviews. (This helps you stay on track when there's a technology snafu, you're tired, or a participant surprises you with comments about her sexual interests. Which really happened to me, by the way.)
- Plan a time budget for each task or topic. This helps you keep interesting (but low-priority) conversations from taking up the whole interview.

Most interview guides consist of an introduction, context questions, tasks, follow-up questions, and debrief. Today you can start by writing the introduction and context questions — on Day 3, after the prototype is complete, you can complete the rest of the interview guide.

Part 1: Introduction

When an interview begins, the participant can be a little nervous and unsure about what to expect. The first part of the interview guide is intended to put them at ease, start establishing rapport, give them an idea of what you'll be doing and what's expected of them.

Start by saying something like this (with a big smile):

Thanks for coming in today! We're constantly trying to improve our product, and getting your frank feedback is a really important part of that.

Before we start, thank you for signing the non-disclosure agreement. I'd like to mention two parts of that. First, it's a reminder that what I show you and what we discuss here today are confidential. It also gets your permission for me to record our session, just for our own internal use. That way we can go back and review it later. And, of course, you're free to take a break or leave at any time during the session.

I like to keep these sessions pretty informal. I'm just trying to learn from you today. I'll ask a lot of questions, but I'm not testing you. There are no right or wrong answers.

I'll start this session by asking some background questions. Then I'll show you some things we're working on, and ask you to do some tasks. As you work on the tasks, please think aloud. This means that you should try to give a running commentary on what you're doing as you work through the tasks. Tell me what you're trying to do and how you think you can do it. If you get confused or don't understand something, please tell me. If you see things you like, tell me that too.

Since I didn't design this, you won't hurt my feelings or flatter me. In fact, frank, candid feedback is the most helpful.

Again… I'm not testing you. I'm testing this product. If you get stuck or confused, it's not your fault. It helps us identify the problems in the product that we need to fix.

If and when you do get stuck, I'm going to try not to answer your questions or tell you what to do. I'm just trying to see what you would do if you were using it on your own. But don't worry — I'll help you if you get completely stuck.

Today we're going to use a prototype. That means some links or buttons or features may not work quite right. You can still click anywhere you like to do the tasks. When you run into something that's not working, I'll let you know.

Do you have any questions before we begin?

Part 2: Context questions

Fight the urge to jump right to questions about the product or prototype you're testing. It's best to start by asking the participant some general, open-ended questions about their past experiences and current behavior. A little context

about their life, habits, attitudes, and problems will help you better understand their reactions and feedback.

You can use personal details shared by the participant to personalize the scenario and tasks when you move on to testing the prototype.

And these initial questions are a great opportunity to build rapport. If you warm up with some easy questions, your participants will feel more comfortable giving you real reactions and honest feedback later in the interview.

When I interviewed potential users of FitStar, a mobile fitness app, I started with these context questions:

What kind of work do you do?

For how long have you been doing that?

What kinds of things do you like to do when you're not working?

What do you do to take care of yourself? To stay in shape? To stay active?

Activities? Sports, exercise, classes? Other habits in your day?

Have you used any apps or websites or other programs to help you with fitness?

Which ones? What did you want them to do for you? What do you like/dislike about them? Did you pay for them? Why? Why not?

Who (e.g. friends, coaches, teachers?) helps keep you active?

How do they help you?

Do you share info about your workouts or your goals with anyone?

When? Why? How?

What (if anything) do you do to keep track of what you're doing?

How does that help you?

How have your exercise habits changed over time?

What did you used to do 6 months ago?

Software and tools you use?

If you had three wishes for things that would make you a more efficient or more effective at getting or staying in shape, what would you wish for?

Let's pretend that for the new year, you want to get more into shape.

Let's start in the app store…

Please go ahead and try to find an app that you'd want to try? [think aloud]

What kinds of things do you look for or avoid?

Format? Types of workouts? Featured celebrities or personalities? Company name?

What do you want to know about these apps before you'd try them?

As in the example above, it's best to start broad, then move on to specific questions related to your goals for the research sprint. If you do it right, participants won't realize the interview has started — it will feel just like natural small talk.

Your interviews are only two days away! By now, you've selected participants and started writing your interview guide. On Day 3, you'll finalize the schedule and wrap up your interview guide. Then you're ready for showtime. Let us know if you have any questions — tweet us at @GVDesignTeam or @mmargolis.

GV research sprint series

Previous: Start recruiting participants (day 1) Next: Finalize schedule and complete interview guide (day 3)

8

How To Decide What Ideas To Prototype | Co.Design | business + design

[Editor's note: This is the fifth in a series of seven posts on running your own Google Ventures design sprint. Read the first part here, the second here, the third here, and the fourth here.]

At the Google Ventures Design Studio, we have a five-day process for taking a product or feature from design through prototyping and testing. We call it a product design sprint. This is the fifth in a series of seven posts on running your own design sprint.

At this point in a design sprint, you've got a lot of ideas down on paper. You've explored the problem, generated a ton of solutions, and looked around at how other companies are solving similar problems.

It's awesome to have a lot of ideas. It's a great feeling. But I've got bad news: You can't build and test everything. And even if you could, it wouldn't be very useful, because you'd have too much information to sift through. So you've got tough decisions to make: Which solutions will you pursue and which will you put on ice?

Today we'll look at how to decide which solutions to flesh out, and how you'll fit them together into something you can rapidly test with users to learn what's working and what isn't.

The decision-making process is hard, and this is one place where working as a group can become a liability. Companies and teams have a natural way that they make decisions—but in a sprint, the group effect can cause decision-makers to behave more democratically than they do in real life. Once the sprint is over and that rosy democratic feeling wears off, you can be left with something that doesn't have true support from the deciders.

To combat this effect, the facilitator often has to draw out the decision-maker to give their honest, true opinion. You'd be surprised at how often this reluctant decision-maker is the CEO. No way, really? Yes way. In the sprint, people are out of their comfort zones, and even CEOs can begin to behave in non-standard ways.

If you aren't conscientious about bringing the decider in now, you'll have a problem later.

One method is giving "super votes" to the deciders during design critiques, which you did in day 2. But most of the time, there are no special techniques. You just have to be blunt.

As facilitator, you should be upfront with the team if you sense you've got a case of groupthink. Let everyone know that you need more assertive participation from the deciders. Additionally, you should have the words "make the call, Sally" on the tip of your tongue throughout day 3. (I'm assuming your CEO's name is Sally.) Don't worry about being a sycophant. If you aren't conscientious about bringing the decider in now, you'll have a problem later.

The first thing I like to do in this phase of a sprint is comb through storyboards from the previous day looking for conflicts. A conflict is a place where there are two or more different approaches to solving the same problem. Conflicting approaches are super helpful, because they illuminate the choices for your product.

For example, let's say you're designing your homepage, where you explain your product to potential customers. Maybe one person's storyboard uses a video, and another uses diagrams on a long scrolling page, and a third uses a single image of the product. Great, you've found a conflict! Every time you find one, write it down. I like to put the topic and solutions on sticky notes, like this:

Each conflict is like a little gold mine. In business-as-usual design, designers often end up picking one approach and going straight to high resolution. When I was working on products in-house, I'd often get so caught up in that one solution that I wouldn't even have time to think about how else it could be done.

So one of the best things about the product design sprint is that it allows you to map out those decision points, and perhaps even to explore a few conflicting ideas in parallel, instead of immediately committing to a safe choice.

You have two basic options for what kind of user study you're going to run at the end of your sprint. You can prototype several different approaches and test them against one another (the "battle royale") or you can go with a single prototype (the "best shot").

The advantage of the "best shot" approach is that you can put a lot more work into that one prototype, or just get it done faster. If you're testing only one solution, the user study is less complex, and it gives you more time to see what the users say about your competitors' products (or just interview users, which is always surprisingly valuable for teams).

Conflicting approaches are super helpful, because they illuminate the choices for your product.

The "battle royale" works well for newer spaces where there really aren't many conventions, and you need to figure which one is going to work best for

the user. The disadvantage is that it takes more time, and your testers may run out of patience before you get all of the information you'd like to have from them. You may have to bring in more participants and run more studies.

On the upside, the results of a "battle royale" can be very surprising. When working with startups, I've often seen a dark-horse design turn out to be the strongest in user studies. When that happens, we thank our lucky stars we didn't "best shot" it, or we never would have known. You may also do some kind of hybrid. Occasionally, if you choose the "best shot" approach, you'll get into testing and find that something's really not working in your prototype, and you need to go back and have a "battle royale" over that specific feature.

So how do you know which to pick? Start with a gut check: If everyone is excited about one option, you may be ready for a "best shot." But if it feels more like you're sitting there and scratching your heads about what to do next—or else you want to throttle each other because those fools just won't agree with you—well, you may need a "battle royale."

What else should you test in your user study? Listing out your underlying assumptions is a good way to revisit the big picture, especially when you've been heads down in a sprint for a few days.

Some of those assumptions might be about the users (example: "Users are willing to upload a profile photo"), some about the business ("the designer-with-glasses-and-beard market is large enough to support our product"), some about technology ("we can automatically cluster profile photos by beard shape"), and maybe even some about messaging ("people will still find beard jokes to be amusing, even for the third time in a single paragraph").

I can tell you that last assumption is false right now, but for most others, you're going to need some kind of research. Guess what? You can test a bunch of them by showing a prototype to users.

For example, if you have a big assumption that users will be comfortable sharing private data in your product, you may want to pick the most aggressive sharing defaults you can think of to prototype. When you show it to users, you'll find out pretty quickly whether your assumption was correct.

Try to come up with a way to test all your assumptions, either in the user study or in some other parallel task that can start right away (e.g., ask the engineers to spend a few hours hacking at that beard-clustering algorithm). If you can't test every assumption now, keep a list for next time. Untested assumptions are like takeout containers in your fridge: If you leave them for very long, things get nasty.

OK, you've picked which conflicts to explore and you've decided which assumptions to test. Congratulations—you're ready to script your prototype.

Now we're going to make a storyboard that shows exactly how the user will step through your prototype, click by click. This storyboard will become the spec for building the prototype. This is an activity that the group does together—it's actually the last group step before you break for prototyping.

Start by drawing a big grid on the whiteboard—each cell should be about as large as two sheets of copy paper, and for most sprints, you'll cover one or two whole whiteboards with your grid. The idea is to draw a comic book that tells a story starting when the user opens the prototype and ending when they complete all necessary tasks.

In each comic book frame, you'll draw a single action—whether it's a pointer clicking on a button, text being entered, or a stick-figure user doing something in real life. You don't have to worry about layout or design in great detail, but you do have to think through every action that takes place in the story.

Drawing the storyboard is hard work, and you'll want to facilitate carefully. Get one person to draw, but don't make them figure everything out on their own. The group should be engaged and discussing what happens next and giving that brave soul holding the whiteboard marker as much help as possible.

When you begin drawing, imagine you're framing the prototype for your user study participant. How will they get to your product? What will they be trying to do when they get there? That'll help you figure out whether the first frame of your comic book is an email or a Google search or an advertisement or the App Store or whatever—and hopefully the story will flow easily from there, following the outline you laid out in day 1.

As you storyboard, there will be lots of small decisions to make that didn't come up earlier in the day. That's expected, since you're working at a finer level of detail now. The facilitator has to work hard here to not let people be too nice. You don't want design by committee. If there's a good argument going, don't try to find middle ground or make people agree. Help the team place a bet on one of the opposing solutions and keep the other in your back pocket if it fails. Call on the CEO to make a tough call when needed. If both solutions are viable, you may want to opt for a "battle royale"—just don't use it as an excuse to avoid decisions.

When you're finished with the user story, take a moment to pat yourselves on the back and eat some chocolate because it's probably been a pretty epic task. You've given form to everything you want your user study participant to experience, and you're ready to turn that story into a higher-resolution mockup.

In the next post, we'll move on to prototyping. It's time for the Fellowship of the Sprint to break up, at least temporarily, as everybody puts on their headphones and cranks out a crafty imitation of a real product. Stay tuned.

- How To Conduct Your Own Google Ventures Design Sprint

- From Google Ventures, The 6 Ingredients You Need To Run A Design Sprint
- The First Step In A Design Challenge: Build Team Understanding
- The 8 Steps To Creating A Great Storyboard

[ILLUSTRATION: Doodles via Shutterstock]

9

The GV research sprint: Start recruiting participants (day 1) | Google Ventures

At Google Ventures, we have a four-day process for answering questions and testing assumptions without the time or expense of launching. We call it a GV research sprint. This is the second in a series of five articles on running your own research sprint. (You can also watch a 90-minute video about research sprints.)

Research sprint checklist:

- Create a recruiting screener
- Post recruiting screener where the right people will see it
- Select and schedule participants
- Start creating interview guide
- Confirm participants
- Complete interview guide
- Review prototype with your team
- Set up test devices and recording system
- Interview five customers!
- Summarize findings and plan next steps with your team

Today you're going to start the process of finding customers to interview. It's not as hard as you might think. Most of the time, it's as simple as posting an ad on Craigslist and screening applicants with a simple questionnaire. I'll show you how to do it the easy way — and I'll also explain how to do it when things get tricky.

Set a tight deadline

The first step is picking a day for your interviews — yes, even before you've finished the prototype or know exactly what you're going to test! (One secret benefit of research sprints is that the deadline is very motivating to your team.)

Find a day when the team will be available to watch. It's important that the core team (designers, engineers, and PMs) observe the interviews and talk about what they learned, so find a day when everyone's available and block it off on their calendars.

Your interview day ("day 4" in this series) should be at least three business days from now. It takes a few days to recruit, so don't wait until you needed the research done yesterday. And try to avoid Mondays and days right before or after holidays; participants (and teammates) are less likely to show up.

Talk to the right people

To get the most bang for your buck, you'll need to define and select the participants you want — and explicitly screen out the ones that won't provide you helpful feedback. While getting feedback from your friends and family is better than nothing (sometimes), you'll quickly see that reactions from "real" users will be much more eye-opening.

When I interviewed coffee drinkers and tested prototypes for Blue Bottle Coffee, I needed feedback from foodies who drink specialty coffee and had ordered coffee online. If I had found out during an interview that any of my participants drank only decaf, or owned a cafe, or didn't brew their own coffee, those interviews would have been a bust.

That's just one example. Every scenario calls for a different recruiting approach, for example:

- Test a getting-started experience with people from the target audience who haven't used your product.
- Test engagement or activation with users who signed up but don't actively use your product.
- Test task completion or usability with people who use (or would use) the feature you are designing.

This is a critical step in the process — one that many teams overlook! Take the time to seriously consider which research participants will help you answer your questions before you start recruiting.

Step 1: Create a screener questionnaire

To find the right people for your interviews, you'll need a screener questionnaire. Every potential participant fills it out, and you can sort through the responses until you find five people that fit the bill. Here's how to create the questionnaire.

Define your criteria

With your team, list the characteristics of the people you want to interview. Then figure out precise criteria you can use to identify those people.

To recruit "coffee-drinking foodies" for Blue Bottle Coffee, we identified precise, measurable criteria I could use to screen prospective participants: they drink at least one cup of coffee per day, they read food-related blogs and magazines, they eat at restaurants at least once per week, etc.

In addition to specifying the users you want to talk to, brainstorm characters you don't want to see in your interviews. You'll probably want to exclude folks

who are under 18 years old, work for competitors, or are unusually technical, including engineers, designers, and product managers.

For Blue Bottle Coffee, we excluded people who roast their own beans, don't enjoy a variety of coffees, or drink coffee infrequently.

Use this worksheet for writing a screener

Write screener questions

Next, write questions for every one of your criteria. Like any good survey, it's important to write questions that don't reveal the "right" answers — some people will try to game the survey so they can get your $75 incentive. For example, rather than asking people whether they go to restaurants and read food blogs, I asked:

In a typical week, how many times do you eat out?

Do you regularly read blogs or magazines dedicated to any of the following topics?

- Sports
- Food
- News
- Coffee
- Cocktails
- Parenting
- Gardening
- Cars

Create a form that people can fill out

After you've written questions for all of your criteria, create your screener questionnaire. I always use Google Forms — it's easy to set up, and the responses go right into a Google spreadsheet that I can sort and filter (more on that later).

Tip: Include a multiple-choice question that lists available time slots. This will save you a lot of back-and-forth emailing to schedule participants. I usually schedule five 60-minute interviews, at 9:30am, 11am, 1pm, 2:30pm, and 4pm.

Look at this example research screener form

Step 2: Get people to fill out your screener

Now that you've created a screener questionnaire, you'll need lots of people to fill it out — that gives you a good shot at finding five that fit your criteria. You can hire a recruiting vendor, but I recommend doing it yourself because it's cheaper, gives you more control, and allows you to complete recruiting in just

a few days. Once you have a screener questionnaire, DIY recruiting is as simple as posting a link where the right kinds of people will see it.

Craigslist

If you're recruiting "normal people" who aren't familiar with your company, you can probably find them on Craigslist. Just post an ad in Craigslist's "Et cetera jobs" section with a link to the screener:

$100 usability interviews on 5/29 (Kirkland)

I'm scheduling 60-minute usability interviews in Kirkland on Thursday, May 29. Selected participants who complete the interviews will receive $100 Amazon gift cards. Please complete this short questionnaire.

Participants will: - Be willing to sign our standard confidentiality agreement. - Allow us to record the session. - Not require an assistive device or software to use a computer.

Look at a sample Craigslist ad

Finding friendlies

If you want to interview people who are interested in your company (or at least familiar), you can post a link to the screener on Twitter, Facebook, or anywhere else you have an audience. You can also add a small link ("Want to give us more feedback?") to your product, website, or company emails.

Experts and hard-to-find participants

When recruiting very specific and hard-to-find types of users, you'll need to get creative.

Check with your sales or business-development teams to see if they have lists of potential customers (leads), partners, or contacts they can share. I've used this approach to recruit medical specialists, small-business owners, job candidates, and many others.

You can also look for contacts in professional associations, community groups, student groups, or your personal network. For example, when I needed to interview restaurant managers, I got in touch with the membership director of my local restaurant association.

No matter where you look for contacts, be sure to share the link to your screener questionnaire and make sure that all potential participants fill out the form. This is the best way to make sure you won't waste time interviewing the wrong people.

For additional tips, check out these articles in the GV library:

By now, you should have a good idea of who you want to interview, and responses from potential participants should be rolling in. With a deadline set and recruiting underway, you can move on to writing your interview guide tomorrow. Let us know if you have any questions — tweet us at @GVDesignTeam or @mmargolis.

GV research sprint series

Previous: Introduction to research sprint Next: Schedule participants and draft interview guide (day 2)

10

The 8 Steps To Creating A Great Storyboard | Co.Design | business + design

[Editor's note: This is the fourth in a series of seven posts on running your own Google Ventures design sprint. Read the first part here, the second here, and the third here.]

Editor's Note12/30/13Happy (almost) New Year! We're saying good-bye to 2013 by revisiting some of our favorite stories of the year. Enjoy.

At the Google Ventures Design Studio, we have a five-day process for taking a product or feature from design through prototyping and testing. We call it a product design sprint. In the first two days of the sprint, we've learned about the problem, shared a lot of knowledge, and chosen the challenge we want to tackle in this sprint. It's time to start cranking out solutions. Expect this step to take between two hours and all day.

I call this step "diverge" because when everyone (from the CEO to the marketing manager) is cranking out quick sketches, we tend to get a lot of ideas—and different kinds of ideas. Remember in the Legend of Zelda how the map would light up rooms you had visited as you explored the dungeon? That's what you're doing on Day 2: illuminating all of the possible paths.

Although you're going to be generating ideas, don't think of this as brainstorming—at least not the everybody-is-shouting kind of brainstorming. Instead, everyone in the sprint will be working quietly and individually, often around the same table. The exercises outlined below force you to get ideas out of your head and onto paper, without getting stuck feeling like they have to be finished or perfect.

In my experience, some of the best ideas that come out of sprints were usually around before the sprint started. It's not that they were bad ideas; they just hadn't gotten enough love yet. The sprint gives you a chance to put all solutions on a level playing field. If you don't bring out your pre-existing ideas, you do yourself a disservice.

Because new ideas are so shiny and fresh, the facilitator needs to remind everyone to think old first. There's no need to be embarrassed of that solution you thought of five months ago while eating a burrito or taking a shower.

One problem with business-as-usual-design is that companies get in the habit of going straight to high-fidelity mockups. In a design sprint, we start designing on paper for a number of reasons:

• It's faster. • Everyone can contribute (not just designers). • Nobody gets too attached to the ideas that are generated because they're so quick and rough. We purposefully use thick markers to make sure nothing gets too precious. • Did I mention it's faster?

Run the series of exercises below to guide everyone from note-taking through sketching and sharing. See my earlier post for an exact list of the materials you need. I use my trusty Time Timer so everyone can see how much time is left in each exercise.

When I'm facilitating a sprint, I like to remind everybody that we're not going to share any of the materials until we make storyboards—that's step five of the cycle—and they'll have plenty of time to polish those up. I want to make sure everybody feels loose and knows they're actually going to have plenty of time to work, even though we're keeping time as we go.

In Day 1, you drew a user story diagram. Look at it together as a team. If the user story is more than two steps long (and it probably is), you're going to need to divide it up before you start sketching. This is as simple as finding natural chunks in the story and drawing a box around them, like this:

Now decide which part to focus on first. It usually makes sense to have everybody in the sprint focus on the same part of the problem at once. If you take that approach, you'll do one cycle for each part of the problem, with everybody collaborating on each part as you go.

You can also divide and conquer—everybody picks a piece of the story they're interested in and works on that. This way is usually faster, although it introduces the risk that people don't think about the user story holistically. If you have more than two or three chunks in your story, you might have to divide and conquer, or perhaps decide you're going to focus on a smaller part of the problem for this sprint.

Either way, the facilitator ensures that everybody knows which piece of the user story they're focusing on before you continue.

At this point in the sprint, the whiteboards and walls are probably covered in diagrams, notes, and "how might we" sticky notes. This is your chance to reload that stuff into your brain. Everyone takes a piece of paper and jots down anything they think is useful.

Now you're going to add all the other ideas that are in your head, mix them with the notes you just took, and loosely organize them on paper. The mind map is going to be your "cheat sheet" you can use when you're sketching UI ideas.

If you're not familiar with mind mapping already, I often describe it as writing down everything in your head with no specific formatting; or quiet individual brainstorming. You can write words and connect them or not, you can

draw pictures or not—you basically can't do it wrong. The important thing is that everyone is getting every solution, old and new, out of their head and onto paper at very low fidelity.

Here's an example:

Everybody folds a blank sheet of paper in half four times, then unfolds it, so they get eight panels. Then you have five minutes total to draw eight sketches, one in each panel. Yes, you did the math correctly, that's about 40 seconds per sketch, which is crazy…but it's a great way to crank out variations of ideas quickly. And since these aren't shared with the group, there's no need to worry about making them pretty.

Since you have only 40 seconds for each drawing, you'll need to turn off the self-editing and just get your ideas on paper. Crazy Eights will also help loosen up your creative muscles and make you more productive in subsequent sketching exercises. If you get stuck, try repeating an earlier sketch with a small variation— this type of exploration is useful and it keeps you moving.

For best results, do two rounds of Crazy Eights. On the second round, everyone will have the hang of it. You're scraping the bottom of the barrel, which makes it more painful to come up with new ideas, but often this is where the most interesting solutions come from.

Now you may be thinking, I'm a bit of a hypocrite: Earlier in this post, I said old ideas are best, and now I'm asking you to come up with new ideas. Don't get me wrong, it's OK to fill out your Crazy Eights sheets with old ideas. But new ones are good too—just because old ideas tend to be stronger doesn't mean they always win.

Pro tip: Get the Bit Timer app for your iPhone and set it to 30-second work periods and 10-second rest periods for eight reps, so you don't have to time it yourself. The rest period alarm acts as your 10-second warning to wind down your current sketch. (Crazy Eights are based on the 685 exercise introduced to us by Brynn Evans.)

Now we're going to make that user story diagram more concrete, and we're going to make something that will be shared anonymously and critiqued by the group. The goal is to take the ideas we've generated so far and sketch an actual UI showing how a user would move through this part of the story— where they click, what info they enter, what they think, etc.

Start with a blank sheet of paper, and put three sticky notes on it. Each sticky note is one frame in the storyboard. It's kind of like a comic book that you're going to fill in. Look back at your mind map and your Crazy Eights and find the best ideas. Chances are, you're itching to illustrate at least one of them in more detail. Now you're ready to rock. I ask everybody to draw UI in the three frames of their storyboard showing a progression: first this, then that, then that.

There are three important storyboard rules: • Make it stand alone: Just like a real product, your drawing has to make sense by itself, without you there to pitch it. In the next steps, people will be looking at these, but you won't have a chance to talk about your idea until the end. • Keep it anonymous: Don't write your name on your drawing. You'll want all ideas to start on a level playing field and it can be distracting to know which one was drawn by the CEO. • Give it a name: Come up with a catchy title for your idea. That makes it easier to discuss and compare later. When you finish the storyboards, hang them on the wall with some sticky stuff. Pro tip: Hang them side by side (like an art museum) so people won't have to crowd in too tight to see them.

Give everybody a bunch of dot stickers. Then, without speaking, everybody looks at the different storyboards and puts a sticker on every idea or part of an idea they like. There are no limits to how many stickers you can use, and I don't even prevent people who want to brazenly vote for their own ideas. By the end, you've got a kind of heat map, and some ideas are already standing out.

Next, everybody gathers around the storyboards one at a time. First, people talk about what they liked, then we ask the person who drew it if we missed anything important. Usually the best, most popular ideas are the ones people can understand without an explanation, so the author of the storyboard often doesn't have anything else to add. This process works far better than letting people explain their ideas first—which almost always uses up a lot of time.

Sometimes I like to do this step on a projector, especially if there are a lot of ideas to get through. I'll take photos of each storyboard on my phone, upload them to Dropbox, put them in a Keynote file, then make notes about parts we like with outlines and text labels as we go through on the projector. This is easier for everyone to see, and you have a digital artifact of the ideas for later. The downside is the setup: Count on 15 extra minutes to capture and upload photos.

Once we've looked at all the ideas, everybody gets one or two "special" stickers (which can be the same dot stickers from before with a pen mark on them). These are "super votes" for the ideas you think are the very best. Between the original heat map and these super votes, it's very easy to see which are the strongest concepts.

The super votes offer a unique way to tweak the process to reflect the decision-making structure of your team or company. Does your CEO make all final decisions about the product? If that's the case, be honest about it and give her three super votes and everybody else one. Or maybe it's a UX director or maybe a tandem of product and design who call the shots. The simple rule is to give the deciders extra votes.

By default, this process will be a meritocracy, but that's not always the way companies work, and frankly, consensus can lead to poor design decisions. The last thing you want are decisions that the deciders don't truly support. On some

teams, these may be unwritten rules, so don't be surprised if it feels a bit awkward to bring it up—in the long run, you'll be glad you did.

Now it's back to the first step to start the whole cycle over again. (Don't worry, it gets easier with every repetition.) If you split up the user story last time, it may be time to move on to another chunk. Often when I'm running a sprint I'll realize at this point that our scope was too large, and we should just double down and keep working on the same section. Either way, the end of a cycle is a good time to take a few minutes and carefully decide where to focus next.

Expect a team to be able to do this cycle two or three times in a day before getting burned out. Throw in plenty of breaks and snacks to keep the troops moving.

Stay tuned for the next episode, where we'll talk about how to decide which pieces of these storyboards go into your prototype.

- How To Conduct Your Own Google Ventures Design Sprint

- From Google Ventures, The 6 Ingredients You Need To Run A Design Sprint

- The First Step In A Design Challenge: Build Team Understanding

11

Google Ventures' Secret Mantra For Super-Productive Meetings | Fast Company | Business + Innovation

Always be capturing.

That's Google Ventures' maxim for making the absolute most of meetings, according to the ultra-insightful writeup of Joshua Porter, HubSpot's director of UX, who recently worked on a product design sprint—yes, that is a thing, and it is awesome—with Google Ventures Design Studio. While the post doesn't go into the results of the meeting—that would spoil the fun—it more importantly shares the process of how to make conversation a little less ephemeral.

From what Porter describes, the products of a meeting are the artifacts that remain after the conversation's finished. So what do those look like? Porter says:

(It's) about the habit of continuously recording the value from your conversation. For example: If you're talking about a new concept, you should be sketching it as you talk so your team has a shared understanding and an artifact of the conversation. We've discussed why you should start capturing before the meeting even begins. Once you've commenced, then go both analog and digital with these tools.

- Whiteboards: On the wall, on wheels—have whiteboards all over the place and let everybody write all over them.
- Post-its: Little square papers are your friends. Porter says to write down bits of ideas and questions on your Post-its, and then you can stick them to the wall or cluster them later. The awesome part of the phsyical world is that you can organize your notes in three-dimensional space, which could help you to see your conclusions in new ways.
- Smartphones: Take photos of your scribbles with your phone; access them later.
- Cloud storage: Have everybody throw those photos in a shared Box or Dropbox folder at the end of the meeting, allowing anyone to grab them when they need them.

To make the most of your devices, Porter says, write or sketch anything that feels important. When you're comparing two things, make a table. If you're brainstorming, sketch it out. That way you're not only engaging your conceptual sense, but your spatial thinking, too. And while you're in a meeting, keep an eye on all those artifacts you've accumulated—they'll let you know if you're repeating yourself, which you probably are.

A tip for effective meetings: Always be capturing

[Image: Flickr user Tanakawho]

12

The Secret Phrase Top Innovators Use - HBR

13

The First Step In A Design Challenge: Build Team Understanding | Co.Design | business + design

[Editor's note: This is the third post in a seven-part series from Google Ventures Design Studio on how to conduct your own design sprint. Read the first post here; the second here.]

Now that you know when to get your team together for a sprint and how to set one up, it's time to tackle the first day of the sprint: understanding.

Chances are that everyone involved in the sprint has different perspectives on the problem—and different information that might be helpful. The goal of the first day is to encourage everyone to share what they already know and develop a common understanding with the rest of the group. By starting at the beginning (even if some people are already familiar with the problem), it nudges the group into a beginner's mindset and leads to fresh solutions. (This is where an outside facilitator can come in handy: Since they're truly new to the problem, their questions can keep the group in a beginner's mindset.)

We use the exercises below to get a ton of information on the table and quickly build understanding. You can do them in any order you want—and even omit the ones you don't find valuable. Try capping each presentation or discussion at 10 minutes. This will keep the day moving and help everyone pay attention.

During the exercises, everyone in the sprint should be jotting down questions on sticky notes. We use the "how might we" format to capture opportunities that might be interesting to explore. For example, "How might we build trust?" or "How might we figure out the user's style?" Often, these end up being extremely useful in the next steps of the sprint.

Business opportunity: The CEO or product leader should walk the sprint team through the business opportunity and market.

Lightning demos: Look at competitors' products. It can also be helpful to look at non-competitive products that solve a similar kind of problem in a different market.

Lay it out: Print out all the important screens in your product, lay it out, and walk through it as a user would.

Success metrics: How will you measure the success of this design? Now's a great time to talk about success metrics. We like to use Kerry Rodden's HEART framework.

Existing research: If you have user research for your product, that's awesome, and you should be sure to go over it. If not, you should talk about whatever data you do know about your customers.

Team interviews: Knowledge about the problem is usually distributed across the company. We've found it very useful to go around interviewing people at the company who have specific expertise, whether that's engineering or sales or customer service. (Customer service people often have incredibly valuable information about the problem.)

Analytics: Look at any data you have on feature usage, where customers drop off your site, conversion rates, etc.

If you're reading this and feeling like the exercises will be tough because you don't have enough data, don't worry! That just means you should do some work before the sprint to quickly gather fresh data.

It could be as simple as scheduling a few user studies or deploying a very short survey with questions related to the problem you're going to tackle in the sprint. You can also do team interviews ahead of time—this is especially useful if you've got a lot of people to talk to.

As a group, use your common understanding to collaboratively map out the user story that's important in this sprint. The facilitator (or another volunteer) should stand at the whiteboard and sketch the flow. This doesn't have to be fancy to be useful. Here's an example:

How do you know which user story is most important? It depends on the problem you are solving in the sprint. For example:

• Helping people understand and get started with your product: You probably want to focus on the experience of a user encountering your product for the first time.

• Creating a new product concept: You probably want to look into the future and imagine the value proposition and core features for an engaged user.

• Improving conversion rate from a landing page: You probably want to understand why people land on your page and what their goals are.

This step can be difficult and time-consuming, but it's critical! Getting a visual map on the wall (like the one above) is invaluable for grounding the discussion and keeping everyone on the same page.

You've got a rapidly approaching user study and there's a good chance you can't prototype everything you uncovered, so you'll need to choose which ideas to move forward. Use your story diagram on the whiteboard to frame this discussion.

At the end of the sprint, when you show your prototype to users, what do you hope to learn? What do you need to design and prototype in order to learn those things? What you decide here will set your course for the rest of the sprint.

The facilitator is responsible for keeping these discussions moving quickly (remember the timer). It's also important to ask seemingly obvious questions because the answers usually further everybody's understanding.

Now that you've fleshed out a common understanding of the problem and started to define which part of it you're going to tackle in this sprint, it's time to move on to the next step: rapidly developing as many solutions as possible.

We'll cover that in the next post, so stay tuned!

• How To Conduct Your Own Google Ventures Design Sprint • From Google Ventures, The 6 Ingredients You Need To Run A Design Sprint

[IMAGE: Football Team via Shutterstock]

14

The GV research sprint: a 4-day process for answering important startup questions | Google Ventures

The founders and designers we work with in the Google Ventures portfolio have a lot of important questions. Is there a market for my product? Will my product help people solve their problems? What should I build? The answers to these questions just raise more questions: Do people understand my product? Will people find my product useful? Can people use my product? And so it continues. As we build our businesses, each answer reveals new unknowns.

User research is a fast, reliable way to answer important questions like these. It's the best way to test assumptions without the time or expense of launching. It reduces risk and helps your team work more quickly and more confidently.

And best of all, you can start doing user research in just a few days. We'll show you how.

Research makes great design possible

At Google Ventures, we help startup teams use design to answer questions and make important decisions about their businesses and products. We like to use design sprints, our own battle-tested process for prototyping and testing ideas rapidly. But design sprints depend on research, so we've optimized a research process that provides maximum learning as quickly as possible.

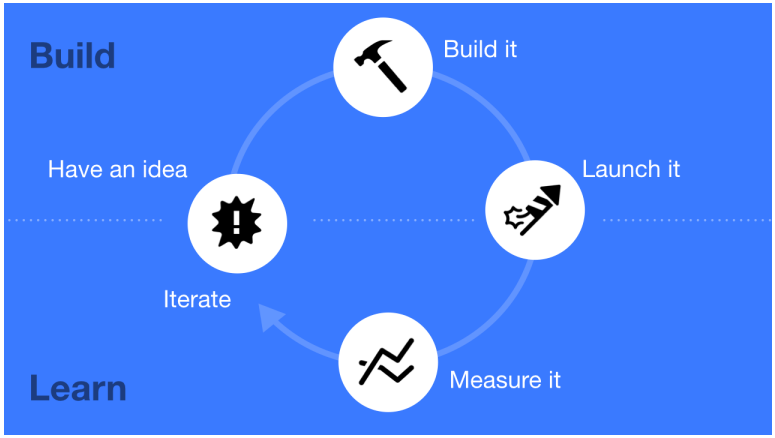
It's called a research sprint, and we've helped more than 100 startups use this process to learn more, make better decisions, and move faster. We've optimized mobile-app registration, improved e-commerce conversion rates, made dense medical data clear, explained new products, and much more.

In the next four articles, we'll provide a complete guide to running your own research sprints.

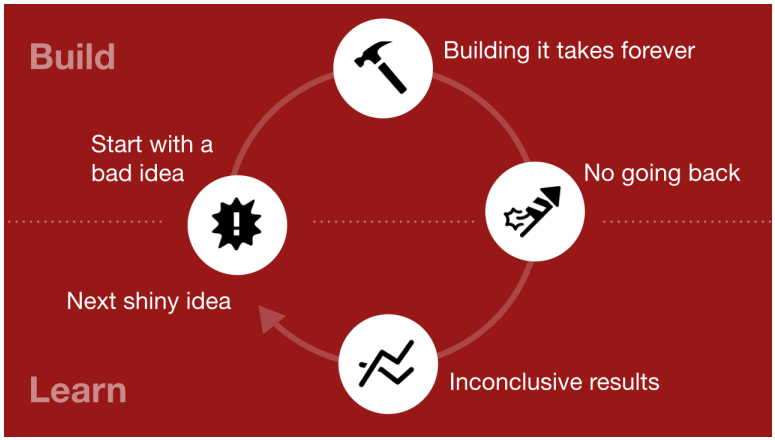
Why research sprints are so important

Most startups know that they don't have all the answers, so they rush to launch things and see how they perform in the real world. This is a very high-fidelity way to answer questions, but it's hard and slow.

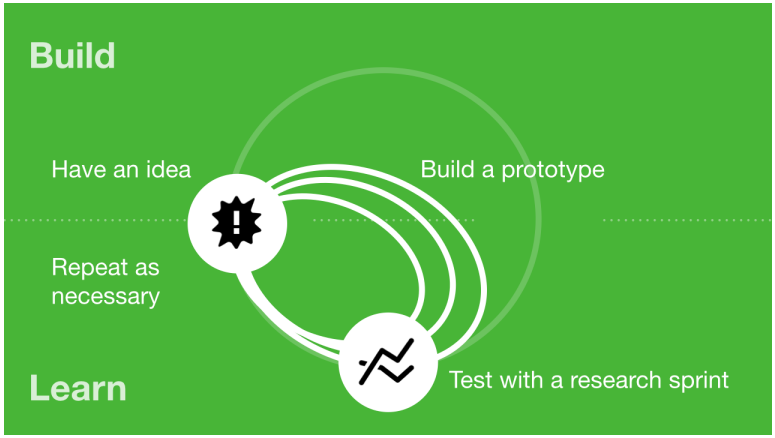
Here's the ideal "launch and iterate" model that's popular in Silicon Valley:



But in practice, it's a tough way to operate. You may start with a bad idea (obviously you don't know it's bad yet). Building it takes longer than expected — it's tough to fix all the bugs, make sure it works in production, and get the details right. It's difficult to "un-launch" a feature, especially once you have people using it. Real-world results are rarely as clear as you hope. And there's always the temptation to move on to the next shiny object instead of diligently iterating on the original idea.



At Google Ventures, we use design sprints to invent and prototype ideas. Research sprints are an essential part of this process — it's how we test our ideas without the time or expense of launching.



In the time it takes to build and launch a new product or feature, you can run several research sprints and improve your prototype after each one. Instead of launching something that might not work, you'll have confidence in the solution you've refined through research.

Then you can invest the time to build and launch your product or feature the right way. It'll reduce risk and save time in the long run.

What can you learn from research sprints?

Research sprints are not a panacea. They won't help you learn how many people clicked a button, where your website traffic is coming from, or how to structure your database. But research sprints can help you answer some of the toughest, most important questions that startups face:

- What problems, needs, and motivations do people have?
- How do people evaluate and adopt products?
- Do people understand your product's value proposition?
- Which messages are most effective at explaining your product?
- Can people figure out how to use your product?
- Why do people stop using your product?
- Why don't people adopt new features when you launch them?

There are many more questions we can answer with research sprints, but I don't want you to get bored and leave.

Components of a research sprint

There are five essential components of a research sprint.

1. A set of questions and assumptions. Without these, you're not making the most of your effort. Before starting the sprint, everyone on the team should agree on the questions you plan to answer and the assumptions you plan to test.

2. Intentional and selective recruiting. You'll need to carefully recruit people based on your goals: existing customers, prospective customers, representative customers, etc. In other words: Unless you're building a product for Starbucks customers, you shouldn't randomly conduct interviews with people at Starbucks.

3. A realistic prototype you can test. You can learn a lot by listening to people, but you can learn way more by seeing how they react to a realistic prototype. The more realistic, the better — you want people's real reactions to what you're building, not their abstract thoughts or smart-sounding feedback. (Check out some real prototypes we built in one day each.)

4. Five 1-on-1 interviews combining broad discovery questions with task-based evaluation of a prototype. 1-on-1 interviews (in person or remotely) are the best-bang-for-your-buck type of qualitative research. You'll learn from facial expressions, gut reactions, and body language. You can ask follow-up questions and follow interesting tangents. Why five? It's easy to spot patterns, and you can do five 60-minute interviews in one day. (Plus, Jakob Nielsen approves.) We'll help you write an interview guide that keeps you on track during the interviews.

5. Real-time summarization of findings. One thing that slows down many forms of research is the analysis — it can take days or weeks to extract findings from the data. In a research sprint, the entire team watches the interviews, takes notes, summarizes the findings, and decides on next steps before heading home for the day.

The four days

As promised, a research sprint takes just four days. Here's what we do on each day:

Day 1 (~4 hours)

- Create a recruiting screener.
- Post recruiting screener where the right people will see it.

Day 2 (~4 hours)

- Select and schedule participants.
- Start creating interview guide.

Day 3 (~4 hours)

- Confirm participants.
- Complete interview guide.
- Review prototype with your team.

Day 4 (~8 hours)

- Set up test devices and recording system.
- Interview five customers!
- Summarize findings and plan next steps with your team.

Let's go!

In the next four articles, we'll explain in detail how to recruit and schedule research participants, create an interview guide, set up a lightweight research lab, and interview customers. You'll be running your own research sprint in no time. Stay tuned!

15

Google Ventures: Your Design Team Needs A War Room. Here's How To Set One Up | Co.Design | business + design

In the last two years at Google Ventures, I've done design sprints with more than 80 startups. One of the simplest tricks I've learned is that a dedicated space with walls—a war room—always helps us do better work. The walls of a war room can extend a team's memory, provide a canvas for shared note-taking, and act as long-term storage for works in progress.

Happy Labor Day! We're taking the day off to fry at the beach and catch up on our Pädagogisches Skizzenbuch. Until Tuesday, enjoy some of our favorite stories from the archives.-Eds.

Unfortunately, war rooms are few and far between. I'm surprised by how many tech companies make space for a foosball table (fun but seldom used), yet don't dedicate a room to their most important project.

If your team doesn't have a war room, don't worry. In this post, I'll explain how to put one together on almost any budget. Spoiler: while a dedicated physical space is great to have, it's not an absolute necessity. But first, here's a bit more on why war rooms work so well.

To solve a complex design problem, you need to track lots of moving parts. As humans, our short-term memory is not all that good—but our spatial memory is awesome. Plaster a room with notes and you take advantage of that spatial memory. You begin to know where information is, which extends your ability to remember things.

We all know it's better to re-order a prioritized list of sticky notes or re-draw a diagram than to make the same decisions verbally. That's why there are whiteboards in meeting rooms and why people love agile trackers with sticky notes. War rooms take those tools to the next level.

War rooms help your team work better together. When you capture every decision and put it on the wall, you don't have to wonder if everyone is on the same page. The room is the page. The more you put on the walls, the more shared understanding you build. As a bonus, you spend less time revisiting already-discussed issues. A war room works great for long-term projects of a few days or a few weeks—and it also works great for one-off meetings.

In a Google Ventures design sprint, it's common to have many things on the walls at once: user story diagrams, research notes, printouts of the existing UI, sketches of possible solutions, a detailed storyboard, and sometimes more. To

accommodate all that stuff, you need a lot of space. That means whiteboards, windows, and empty walls where you can stick stuff.

Every bit of window, wall, and whiteboard is useful.

You don't want your war room turning into just another conference room. For best results, remove your war room from your company's room-scheduling calendar.

Whiteboards come in a lot of styles, so choose wisely.

- Floor-to-ceiling wall-mounted—The best. I like to use every square inch of available space, and with these babies, that's a lot of space.
- IdeaPaint—Great stuff (unless your walls have a funky texture). And for goodness sake, paint all the walls, otherwise, get ready to have somebody write "Not a whiteboard!" in whiteboard marker on the unpainted walls.
- Normal wall-mounted—These are okay if you get more than one.
- D.I.Y. shower board whiteboards—Much cheaper than real whiteboards, these require more elbow grease to install (you may spill Liquid Nails on your designer-y plaid shirt). The surface isn't quite as good, so expect to clean it more often.
- Rolling—These come in small and giant sizes. The small ones have a lot of unusable space down by the floor, and they shake when you draw on them. The giant ones cost a lot more, but they're actually usable.

In our design sprints, we go through a lot of different work modes. Sometimes we need to talk a lot, and we want chairs and open space. Other times, we're drawing on paper and we want desks. The ideal war room has furniture that's lightweight or on wheels, so it's easy to move.

Everything is lightweight, on wheels, or both. You should always have at least one person wearing plaid—three or more if possible.

We took over a conference room and removed the big table in the middle. Next, we installed as many whiteboards as we could. We couldn't do floor-to-ceiling, but we got close.

Finally we ordered a bunch of flexible furniture—some of it fancy-pants (like Modernica chairs) and some utilitarian (like clipboards and a coat hanger). Here's the complete shopping list hand-picked by Google Ventures' Daniel Burka. Some highlights:

It may be impossible to completely take over a room. If you have to share your war room, get some portable wall space that you can assemble and disassemble quickly. Your options:

- Sticky flip charts—Blank sheets of this stuff make a reusable, moveable backdrop for sticky notes and printouts.
- Giant foam core—Foam core comes in 96"x48" but it's expensive and tricky to find, not to mention cumbersome. Which is why I prefer...
- Rolling whiteboards—see above for our favorite.

Sometimes you don't even have a conference room to commandeer. I've seen this challenge at startups in incubators or shared offices. Don't freak out. You can still make a war room by hacking the space around your desk. Use rolling whiteboards as partitions. It's just like you're a kid again, building a fort out of chairs and blankets! But don't actually use blankets, because your co-workers might get creeped out.

We're still experimenting and learning with our own war room, as well as those at our companies. How have you set up project spaces for your team?

16

From Google Ventures, The 6 Ingredients You Need To Run A Design Sprint | Co.Design | business + design

[Editor's note: This is the second post in a seven-part guide on how to conduct your own Google Ventures' five-day design sprint. Read the first part, on why you should conduct a sprint, here. See more at Google Ventures's site, Design Staff].

At the Google Ventures Design Studio, we have a five-day process for taking a product or feature from design through prototyping and testing. We call it a product design sprint. This is the second in a series of seven posts on running your own design sprint.

Now that you know what design sprints are good for, you'll need a few important ingredients to make yours successful. Start with a big, important problem; pitch it to your team; and schedule a user study before you even start. Get the right people and the right supplies in a room and you're on your way to a successful design sprint.

The first thing you need is an important design problem, and if you work at a startup, chances are good you probably have one lying around the office. Maybe more than one. It might be something big, like defining your product for the first time, or a big redesign or new feature. Or it might be something detailed, like improving conversion on a single user action. It just has to be really important to the company, and it has to be something you're struggling to start or to make progress on—otherwise it can be difficult to get the other people you'll need involved.

As long as it's an important problem, it's perfect for a design sprint. It's OK if you don't feel ready to start on it yet. No matter how overwhelming or ambiguous, you'll be able to cut a big swath through the jungle of possible solutions.

The ideal sprint team is between four and eight people, but you can get by with more or fewer than that. Just make sure you have at least one:

• Designer: If your startup doesn't have a designer yet, try to bring in a ringer.

• CEO: At a small startup, the CEO is the key decision-maker and needs to participate in order to get an actionable solution out of the sprint. At a bigger company, you'll still need buy-in and it's best to include the CEO, but if they

can't be there the whole time, you can bring them in at key decision-making moments.

• Product manager: The PM (or whoever is filling this role) will likely need to implement the solution that comes out of the sprint.

• User expert: The person on the team who has the most direct contact with customers often has great input, and can be the lead on user testing.

It's also great to include: • Engineer • Marketer • Anybody else who's interested

Once you know when you're going to do the sprint, recruit users and schedule the user studies for the last day of the sprint. This is a bit terrifying: you haven't even started to talk about the problem, let alone design solutions, and people—outsiders!—are going to come in and need to be shown something. This hard deadline, even though it's artificial, will help you move faster and make tough decisions to focus your work throughout the sprint.

Pick someone to be the facilitator of the sprint. The facilitator is going to be responsible for managing the sprint and moving things along. They need to be confident leading a meeting, including synthesizing discussions and telling people it's time to stop talking. They also need to be comfortable with not getting to participate as much in the actual design work, because facilitating is a lot of work. Since you're the one reading this, you may be a good candidate—but it's always easiest if the facilitator is an outsider. See if you can get a friend from another company to help out.

Clear everybody's schedule for five consecutive days. It's also very important to have a dedicated room for the duration of the sprint, usually a conference room with lots of whiteboards.

Much of the magic in design sprints comes from the sense of urgency. By their very nature, startups always feel time-constrained; the short, focused time of the sprint adds another constraint.

Luckily, you don't need anything fancy to run a sprint. Here's everything I use:

• Sticky notes: I like the yellow 3×5 size. • Drawing pens: Any standard black or blue pen is probably fine. I like these, or you can get these for extra credit. • Whiteboards: If your war room doesn't have a lot of whiteboard space in it, find another war room or some rolling whiteboards, or heck, get some IdeaPaint and get busy. • Whiteboard markers: I like to use these instead of Sharpies because they're so versatile. Buy some good ones and be sure to have enough black markers for each person in the sprint. • Dot stickers: for voting. You want something small with uniform color. Post-It brand dots are great. • 8.5 × 11 blank copy paper: Nothing special, just have a pack of this on hand. • Time Timer

Clock: Optional, but totally awesome, see here. I guarantee you'll find it useful during the sprint, and probably during regular meetings afterward. • Snacks: You'll need caffeine and food handy. Trail mix, bananas, and dark chocolate covered raisins have proved especially popular at our sprints, although it is possible that it's just me eating all of it. • Sticky stuff: You'll need to stick your drawings and storyboards on the wall. This removable gummy material is inexpensive and works great, with less fuss than tape.

OK, the stage is set. Now it's time to start the sprint.

Stay tuned for tomorrow's post, in which we'll explore the exercises we use on the first day of a sprint.