

 workingmouse

**Your Vision,
Our Expertise.**

Activity Kit

Stage 1 Discovery
2023-02-17

Contents

-  [Activity Kit](#)
-  [Techivities](#)
-  [Data and Feature Security](#)
-  [Estimating Backlogs](#)
-  [Information Architecture](#)
-  [Paid Services](#)
-  [T-Shirt Sizes](#)
-  [Target Platforms](#)
-  [Tech Spikes](#)
-  [Third Party Integrations](#)
-  [Writing a Backlog](#)
-  [Designivities](#)
-  [Lean UX Canvas](#)
-  [5 Love Languages](#)
-  [Assumptions List](#)
-  [Breadboarding](#)
-  [Crazy 8's](#)
-  [Custom Components](#)
-  [Design Principles](#)
-  [Discovery Interviews](#)
-  [Icons](#)
-  [Jobs to be Done](#)
-  [Lean UX Canvas](#)
-  [Legacy system demonstration](#)
-  [Motivation Card Sorting](#)
-  [Personas](#)
-  [Project Focus](#)
-  [Red Routes](#)
-  [Screen Resolution](#)
-  [Trade-off sliders](#)
-  [User Flows](#)
-  [User Story Map](#)
-  [User Testing](#)
-  [Warm-ups](#)
-  [Processtivities](#)
-  [Async huddles](#)
-  [Retrospective](#)
-  [Roadmap](#)



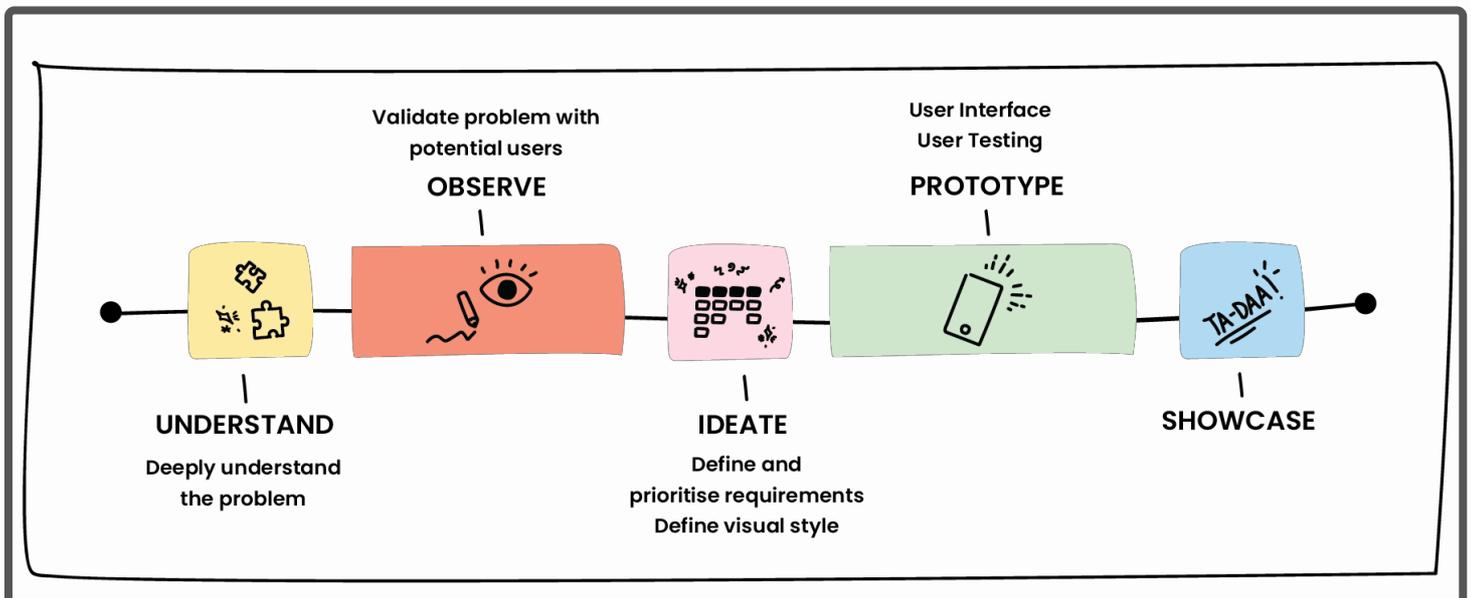
Activity Kit

Introducing the **WorkingMouse Activity Kit** – a powerful collection of tools and resources, designed to tackle any challenge that comes your way over the course of a project. Carefully curated by industry leaders and tested in real-world projects, this document is your secret weapon for success. Imagine having a blueprint for solving problems, with clear instructions on how to perform each activity and the desired outcomes.

In stage 1 discovery, we use the activity kit as a tool to better understand and investigate our customer's problems by selecting the most relevant activities and running them with the team. By doing so, we can gain valuable insights into our customers' needs, pain points, and behaviors, which will inform our product development and design decisions.

Think of the Activity Kit as your toolkit for software development, aided by design. It is loaded with high-value activities like crafting compelling problem statements, mapping user stories, conducting discovery interviews, creating a product backlog, prioritizing projects, running retrospectives, and facilitating user testing sessions. The activities listed within this kit are used throughout our scoping phase, which is twofold; to thoroughly explore the problem statement and design a solution, and to prepare that solution for development. The scoping phase can be broken down into five specific phases:

1. **Understand** the customer and their needs.
2. **Observe** and gather insights through user testing to confirm the problem statement.
3. **Ideate** and generate various solutions to the problem.
4. **Prototype** the strongest solution and gather feedback through testing and adjusting as needed.
5. **Showcase** the final product by completing the requirements backlog and aligning the database model with the validated prototype.





Techivities

Techivities are an innovative set of activities that are specifically designed to supercharge the technical aspects of software development. They are used throughout the development process to deliver high-quality software. Unlike traditional approaches that focus on general software tasks and the development process, Techivities are tailored to the specific technical challenges of a software development project, providing targeted support that can help to ensure success.

Techivities include activities such as tech spikes, where developers can explore new technologies and assess their potential impact on the project, and estimating backlogs, which helps to ensure that project timelines and budgets are met. They also provide guidance on how to handle technical debt, maintainability, scalability and performance of the software to deliver high-performing, stable and scalable software. With Techivities, software development teams can achieve new heights of technical excellence and deliver outstanding results.



Data and Feature Security

This investigates the data access risk, and assists with the investigation of the data and feature security issues to be identified including highlighting the development risk associated with that. You as a developer should work to determine what data and features can be accessed and by who. This activity assists with implementing those requirements and enforcing the principle of least privilege such that access is not possible unless specifically allowed.

Activity Details

Activity Lead

Developer

Who's typically involved?

Developer

No. Participants

1

Est. Completion Time

2hrs+

Steps

1. Locate the Tectivity - Data and Feature Security document in your project space
 2. Complete the templates:
 1.  Access control by user group
 2.  Risk assessment (Incorrect access)
 3.  Segregation types
-

Template

Access control by user group

Features

Feature	User group 1	User group 2	User group 3	User group 4	User group 5
Feature 1	X	X	-	-	-
Feature 2	X	X	X	-	-
Feature 3	X	X	X	-	-
Feature 4	X	X	-	-	-

Risk assessment (Incorrect access)

If the data was:

- Accessible / available to the wrong user(s)
- Lost / unrecoverable
- Breached / hacked

Risks

Low

Impact: on the business

Actions (recommended or required)

- No additional action required

Medium

Impact: violate a client's policies, such as privacy policy or terms and conditions, result in legal action

Actions (recommended or required)

- Data recovery meets client requirements
- Occasional penetration testing

High

Impact: violate government policy, result in significant legal action

Actions (recommended or required)

- Data recovery meets client requirements
- Regular penetration testing (3rd party)
- Regular audits (3rd party)



Segregation types

If the data was:

- Accessible / available to the wrong user(s)
- Lost / unrecoverable
- Breached / hacked

No data segregation

< Add detail here > E.g. This is the case where the entire model is either available or not available. There may be some simple exceptions determining the access, but for the most part all entities have the same access.

This case is unlikely to need any custom security code. (Depending on the feature access control type)

Examples:

- A public library, where everything is available to view.
- A private library, where everything is available, but only if the user is logged in. Otherwise, nothing is available.

Per Entity

< Add detail here > E.g. This is the case where individual entities are available in some conditions while others are not, but it is the whole entity and all data for that entity.

The security diagram is usually enough to handle this case. (Depending on the feature access control type)

Examples:

- An entity for keeping an audit trail, which only admins should be able to view.
- A news website, which allows any user to read the articles, but only allows a user to comment if they're logged in.

Per Column

< Add detail here > E.g. This is the case where access to columns on the entity are limited, such that a user can only CRUD a subset of columns.

Preventing a user from reading data in a specific column can often be handled by the security diagram, but preventing them from saving data back into that field will usually require custom security code. However, a baseline Codebots app will already have a few Per Column hidden fields, which shouldn't need extra attention.

Examples:

- An entity which has some columns which are public while others are limited to logged in users.
- A user column for saving hashed passwords, which shouldn't be fetched from the front end.
- An online learning website, where a unit's details are public, but the HTML of the page is hidden until the user is logged in.

Per Row

< Add detail here > E.g. This is the case where access to a row or record is determined by a particular column value. This could be as an Enum, or it could be as a relationship to another entity, and it could be on any entity.

Note that an entity could have both Per Column and Per Row filtering. However, Per Row filtering is generally more complex.

Per Row data segregation will usually require custom security code. One possible exception is owner-based security, which the security diagram metamodel can handle. (Although the Codebots platform currently cannot)

Examples:

- An entity which has an Enum column such as "Private" or "Public", and access is only granted to records "Public" when not logged in.
- An entity which is attached to a specific user. The user can only access their own records, and not those of other users.
- An entity which belongs to a specific organisation. The user can only access it if they belong to the same organisation as the entity.

- An entity which belongs to a specific organisation. A user from that organisation can access that entity, but only if they have been given additional access permissions.



Estimating Backlogs

This task calls for the scoping team to work with an additional developer, along with a Squad Lead to estimate the backlog. Estimations are completed using time and a risk factor per ticket and are calculated using an in-house tool. The output will be an approximate timeline of the effort involved to deliver a defined scope.

Activity Details

Activity Lead

Solution architect

Who's typically involved?

Solution architect, software developers, product designer, and squad lead

No. Participants

2-5

Est. Completion Time

2-8 hours (depends on the backlog size)

Template Link

We have an estimations template, which is available upon request.

Before starting, an Estimation Spreadsheet must be created for the project and set up correctly for the estimations at hand.

Setting up the estimations spreadsheet:

1. Export the story map from Miro ...
2. Run the CSV to structure the list of epics, stories and priority statuses
3. Download a copy of the estimations spreadsheet
4. Go to the 'Backlog' worksheet tab

5. Paste program output
6. Schedule in a day to estimate!

Steps

Before the estimation process should start, it's important that the development team have taken every effort to meet the Definition of Ready for every ticket up for estimation. This will enable the session to be as efficient as possible. It is near unavoidable to continue to discover new learnings during estimations, but it's important to make our best effort.

1. The Squad Lead (or Product Developer) should begin by setting up a new estimation spreadsheet and bring in all tickets with appropriate priority, epic link and type.
2. Setup preliminary options for All, Must, Should, Could on Options Tab based on Epic and Priority.
3. Check all variables and allowances suit the project in question.
 - For Delivery projects - confirm the variables for Iteration 0 estimate, beta release estimate, number of Launch Iterations required, length of each Launch Iteration. It is also worth confirming the required Allocation Factor, Discovery Allowance and Delivery Allowance.
 - For all other estimates, all variables and allowances can be removed as the tickets will be treated individually once the estimate has been derived from the time and risk score allocated.
4. Assemble the team in a meeting (for the expected length of the session), have the estimation spreadsheet, backlog and prototype open (and any other supporting materials).
 - It is preferred that the supplementary developer joins the entire session, but in the event that is not possible - they may join just the developer later and estimate on a select sample of tickets.
5. Work through the tickets one by one, explaining the tickets as required for clarity to estimate.
 - New points discovered during the session should be noted on the ticket and added to the relevant section of the ticket.
6. For each ticket, it is typically to estimate each item one by one in the following order:
 - Development Time
 - Testing Time
 - Complexity
 - Unfamiliarity
 - In the event there are large discrepancies at any point, it's recommended to discuss and attempt to re-align. The expectations isn't that the same estimates will come out for each, but if they are largely different there is a risk that the developers have different understandings of the same functionality.

Extra resources

- [Planning Poker: An Agile Estimating and Planning Technique](#)

Tips

- This process scales for small groups of tickets (ignoring any factors) or to a large project
- Typical suggestion of the maximum length within the estimations is between 15-20 weeks of development for 2 developers.
- Estimation poker can be a useful technique to remove bias. There are cards around the office that will allow the developers to count in and present their estimates and risk score.
- A key point where discussions should occur is when the two or more developers involved have a larger than 1 jump between there estimations. Eg. one developer estimated 1 hour, while the other estimated 4 hours.



Information Architecture

The Information Architecture (IA) is a primary deliverable at the completion of a Scope. The purpose of this Technicity is to discuss the architecture of the app, or if changes might be required, and produce an IA page.

Activity Details

Activity Lead

Developer

Who's typically involved?

Developer, DevOps

No. Participants

1

Deliverables

- Project Information Architecture page
 - Deployment Architecture page
 - Documentation outlining any environment and performance considerations
-



Steps

1. Book a meeting with DevOps to discuss the requirements of your app.
2. Document any environment complexities that will be required to be considered for deployment or performance.
3. Produce the IAD page. It should live under "Documentation".
4. Start with our Information Architecture Sample and update it accordingly.
5. If the diagram is not accurate, create a custom diagram.



Paid Services

A breakdown of all of the paid services which we are planning to use in the development of our client's platforms. As well as the purpose of including these services, and their pricing information.

Activity Details

Activity Lead

Developer

Who's typically involved?

Developer

No. Participants

1-2

Est. Completion Time

30 mins

Steps

1. During scope, most likely in weeks 2 or 3, the Product Developer fills this activity out once they have an idea of which third party api's etc they plan to use.
 2. PSSST! We've left all of the services in the template that we've used in the past to give you a head-start. Delete the ones you don't intend to use.
 3. It is critical that this page is presented to the client before scope handover, either in a scope meeting or via email. As we don't want to surprise clients with paid services when they get into the development phase.
-

Tips

- If you're concerned about the cost of some services, talk your clients through what other options they may have, and let them ultimately make the call.
-

Template

Paid Services (v1.0.0)

This page contains a breakdown of all of the paid services which we are planning to utilise in the development of [INSERT PROJECT NAME], as well as the purpose of including these services, and their pricing information.

For Workingmouse to use these tools in development, we will need API keys for each service which will be sent with the requests. It will be the responsibility of the [INSERT CLIENT NAME] team to organise deals/purchase access to these services and provide us with the API keys which we need, prior to development starting.

Address Autocomplete database

Address autocomplete will allow us to display a list of addresses when the user is entering their address into the site. The pricing plans for this service seem very dependent on knowing your expected use and can get very expensive very quickly, so some thought will need to go into this one from the Inspire team to work out which deal they want to go with

Pricing: [Postal address finder pricing and plans](#)

Emails

Instead of using mass marketing email services such as Drip or Mailchimp in this build, we will rely on C#bot's email service to send out basic emails such as 'account confirmation' and 'reset 'password links' etc.

However we will require [INSERT CLIENT NAME]'s existing mail server details to send them from.

Google Maps API

The Google Maps API is used for displaying a map with all of the driving schools in an area within the site. We will need access to the Dynamic Maps API, as it will allow us to drop pins on the map. It appears that google offers a pay as you go system, so we do not need to worry about selecting a plan based on our expected usage

Pricing: [Google maps API](#)

Hosting

Once we have finished development and ready to deploy Licence4Me, we will need to consider how it is going to be hosted so that it is publicly accessible on the internet. I don't have much information on pricing for this at the moment as I have not examined hosting requirements for the site yet, but I am having a meeting to discuss it tomorrow and will be able to provide more information after that.

Payment providers

E-way

E-way is a payment gateway that we can use for integrating payments into the site. It is a more cost effective solution compared to Stripe, however we are aware that its not as performant.

Pricing: [Eway](#)

Stripe

Stripe is a payment gateway that we can use for integrating payments into the site

Pricing: [Stripe](#)

SerpAPI

SerpAPI is used to search google for driving schools and their reviews. In terms of pricing, Inspire will most likely need to purchase a developer subscription for the development process, and then update to a big data/enterprise plan following that.

Pricing: [SerpAPI](#)

SMS messaging providers

Twilio

Twilio is used for sending SMS messages from within the application.

Pricing: [SMS Pricing in Australia for Text Messaging](#) | Twilio (4.9 US cents)

Verification pricing: [Verify Pricing](#) | Twilio (5 US cents + 4.9 US cents = 9.9c)

Firebase (Authentication only)

Firebase is an alternate service we could use for sending SMS messages from within the application. However they only send SMS messages for authentication purposes (giving users access to their account).

Pricing: [Firebase](#)



T-Shirt Sizes

The T-Shirt Sizes activity allows teams to broadly estimate the size of a project during scope. Previously, teams have not been able to give clients an estimate on how long their project is going to take until the end of scope.

Activity Details

Activity Lead

Solution architect

Who's typically involved?

Solution architect and software developers

No. Participants

2-3

Est. Completion Time

2 hours

Template Link

- [T-shirt Estimations \(internal link\)](#)

In many circumstances, this has caused bill-shock for a number of clients. This activity aims to give clients an indication of how big their project is getting. Ideally, this will trigger discussions of project size, budget, and priorities earlier in the scope, enabling the scoping team to manage expectations around final estimations.

Steps

1. Perform the User Story Mapping activity with the Designer and the Client.
2. Once the story map has enough detail such that t-shirt sizing can reasonably occur, you can proceed with this activity. This will typically occur around the middle of the scope time-frame.
3. In Miro, use the tags to apply T-shirt sizes to each user story.
 - Small, Medium, Large.
 - These sizes are based on your judgement in relation to the project (don't go straight in thinking that S must equal 2 hours, therefore M must be 4 hours etc.). In the following steps, you will use sampling to determine the duration of each T-Shirt size.
4. Export these issues from Miro as a CSV and import them into a copy of the T-Shirt estimations spreadsheet template.
 - Add your backlog items into the Backlog Sheet, making sure that the issue name, epic and are correctly set.
5. Once you have added a T-shirt size to every ticket, select your samples and add these into the T-Shirt Samples Sheet.
 - It is recommended that an even number of tickets is selected of each size grouping, 3 of each is a good guide i.e. 3 x S, 3 x M, 3 x L.
 - The issues selected are discretionary and based on the developer's best judgement
 - It is recommended that samples are selected to provide the best coverage, as such, some methods include:
 - Random selection
 - Selecting one high risk, moderate risk and one low risk of each size.
6. Determine the time estimate that each T-Shirt size is equal to. You must apply the same time estimate to a single T-Shirt size. E.g. All three mediums = 6 hours each.
 - 💡 TIP: Start with the M sized samples. Once you have estimated that all three M samples are equal amounts of work, i.e. M = 6 hours, you can now use this as a basis for your S and L estimates.
 - NOTE: Each size is not required nor expected to be a multiple of any other.
7. Define the point values for each size in the Options tab.
 - 💡 TIP: For the best results, attempt to ensure an equal number of hours per point by adjusting the point values in the Options tab for each size.
8. Once you have an estimation for all three T-Shirt sizes, you can apply these estimates to the rest of the user story map. NOTE: This is completed already for you by the supplied spreadsheet template. To do this:
 - Apply risk factors:
 - Multiply the total time frame by 165% to account for risk, discovery and allocation.
 - Apply the multi-developer factor
 - Multiply the number of developers by 0.8 to calculate the multi-developer factor
 - Divide your total single developer time frame by the multi-developer factor to calculate the adjusted timeframe
9. Update the spreadsheet variables if necessary. i.e. number of developers.
10. Read final estimate from spreadsheet



Tips

- It is important to stress that this number has been calculated with the limited information we have at this stage of the scope, in order to give the client the earliest possible indication of scope size. The estimates WILL CHANGE at the end of scope, once the backlog is finalised and the scientific estimations process is performed.
- Use these estimates to guide the reminder of the scope time frame... does the client need to prioritise to cut work out? ... or are they on track to keep their 'nice to haves'?



Target Platforms

This activity should be brought up in the first scope meeting and is designed to briefly identify atypical aspects of the project. By default, we target the most used modern browsers or the newest OS at the time of development. We also test against them, however there are situations where customers have very specific needs for old platforms or specific platforms we wouldn't normally consider.

Activity Details

Activity Lead

Developer

Who's typically involved?

Developer and client

No. Participants

2-3

Est. Completion Time

30 mins

Steps

1. During or after the Scope Kick-off Meeting, the Developer fills this activity out.
 2. Knowing this information will inform the design of the platform's architecture.
-

Tips

- Be cautious of scoping out both web and mobile apps at once. If this occurs, talk to your Account Manager and/or Head of Product to come up with a plan for development.
 - Tip from Chris - its critical to define the platform's load & performance requirements here. How many users and how much expected data is there? If its really large, testing out the performance is an easy win with a tech spike.
-

Template

Target Platforms (v1.0.1)

Questions	Answers	Notes
Is the app for this build targeting Web or Mobile? (not both)	Y/N	...
IF Mobile, is the client aware of Android and Apple Ts&Cs? Some key call-outs: payment/subscriptions, account management, user data, external biometrics	Y/N	If yes) initiate the app stores accounts setup DURING SCOPE. Please refer to the latest version of the Iteration 0 Checklist for most up-to-date instructions. Look for: Mobile / App Store Accounts, Mobile / Apple Bundle Certificates
Is this app to be used in a heavily controlled environment?	Y/N	...
Do your users get to choose their own devices/browsers?	Y/N	If yes) do you know what the large majority use?. If no) what are the allowed devices/ browsers?
Are there specific versions users have to use?	Y/N	If yes) list them
Is this app to be used in a heavily controlled environment?	Y/N	If yes what is the estimated user load?
How much data could you foresee being stored per user? or other relevant metric	Y/N	...
Will this be locally or cloud deployed?	Local/Cloud	If locally) do you have the infrastructure and staff members to deploy and maintain the application? If cloud) do you have a preferred provider?



Tech Spikes

The purpose of a Tech Spike is to de-risk (the complexity and uncertainty of) a technical requirement and document your findings for the development team. This may include complex UI components, interactions between entities, or integrating 3rd party packages etc. All research, findings and documentation should go into the Techspike page, or as child pages under it.

Activity Details

Activity Lead

Developer

Who's typically involved?

Developers

No. Participants

1

Est. Completion Time

4hrs - 1 day

Steps

1. Create a page from the template to document your research
 2. Put it under the tech spike folder in the Gitlab project
 3. Complete the sections -
 4. Overview:
 1. Document the requirement and reason for the tech spike
 5. Investigation Summary:
 1. Add any outcomes as rows which are required to report on
 6. Research:
 1. Record all research and findings, useful links to resources (such as NPM packages, Stack Overflow threads, etc), implementation notes including screenshots where useful
-

Tips

- You should work to determine the resources required, implementation details, and any costs that may be incurred by the client to implement the solution, and physically test these things where possible.
-

Template

Tech spike name

Overview

Requirement

i.e. Define the customers requirement here.

Reason for Investigation

i.e. Define why the requirement has a high level of risk, complexity or unfamiliarity.

Investigation summary

Questions	Answers
Are there any costs involved?	<i>Amount</i>
Does this require a third party integration?	<i>NPM package, API, etc.</i>
What resources are required?	<i>Servers, tools, software</i>
What new skill set(s) are required?	<i>Languages, Domain knowledge</i>
Is this business critical?	<i>Yes/No</i>

For integrations, include answers to the table below. Else delete it

Questions	Answers
Is the integration run on a service outside of WorkingMouse?	<i>Yes/No</i>
If so, what happens if that service goes down?	<i>Details of fallback</i>
How often does the service we're integrating with get updated?	<i>Often/Not often</i>
Would an update require a refactor of our application?	<i>None/Minor/Major</i>

Research

Key Links

- ...

External

e.g. Stack Overflow threads, links to tools, npm packages, youtube tutorials, documentation.

Internal

e.g. Confluence links, relevant internal documentation

Documentation

i.e. experiments, screenshots, key documentation excerpts

Branches

If code was completed to help demonstrate the functionality push this to the client project under a branch techspike/topic then link the git lab page here



Third Party Integrations

One of the major benefits of building bespoke software is the opportunity to integrate disparate systems and tools to maximise the efficiency and reliability of a process or workflow. As such, it is common to integrate with one or more third parties as part of a project. As each third party may operate differently with a different integration path/process, it is important to document each one. The purpose of this activity is to document the specific details surrounding each integration to ensure that the key aspects have been considered.

Activity Details

Activity Lead

Developer

Who's typically involved?

Developer

No. Participants

1

Est. Completion Time

2-4hrs

Steps

1. Fill in the  Third party integrations template with all integrations that you are considering integrating with, it doesn't matter if you don't have all the details yet.
 1. Start adding to this list throughout scope as you hear any mentions of possible integrations. It should be located under the documentation section of your customer space.
 2. Investigate the integration details/path for each integration and link any appropriate supporting documentation.
 3. Create risk items in the Risk register for anything that cannot be filled in.
 4. Raise tech spikes for any integrations that appear are high risk.
-

Tips

- Use tech spikes to validate any assumptions regarding an integration. If you cannot test an integration during scope, it is likely that there will be challenges during development.
 - Some integrations require payment/do not exist, therefore it is important to perform this activity as soon as possible to manage the risks associated with it.
-

Template

Integration type/name	Cost	Open API available?	Up to date documentation available?	Test date available?	Sandbox environment for testing?	Notes
...	...	Y/N	Y/N	Y/N	Y/N	...
...	...	Y/N	Y/N	Y/N	Y/N	...



Writing a Backlog

The Backlog is a primary deliverable at the completion of a Scope. The purpose of the Backlog is to catalogue the functionality of the application into Epics with Stories and Tasks to give the development team a plan as to what needs to be done to deliver a complete application that meets the clients' requirements.

Activity Details

Activity Lead

Solution architect

Who's typically involved?

Solution Architect, product designer, squad lead, and software developers

No. Participants

2-5

Est. Completion Time

3-5 days

Template Link

Use the 'User Story Map' shared template in [Miro](#)

Steps

1. Complete User Story Mapping and User Journey activities with the Designer and the Client. Once this is complete, control of the backlog effectively shifts to the scoping Developer, as they have the technical knowledge of how the stories will be built.
 2. The User Story Map will already be established in Miro and contain Epics & User Stories.
 3. You may choose to finish writing the backlog in Miro (preferred) or write it in Confluence.
 4. For each Story and Task, include the details and context discovered during prior activities.
 5. There will be a number of fields on each ticket that must be completed, read about Backlog components below.
 6. Review the Stories and identify any that need to be separated into multiple Stories or Tasks. Ultimately, tickets should be broken down into the smallest possible size for it to be developed, released and deliver value on its own.
 7. Complete any implementation and test plan details for each story and task these are suggestions and the developer building the application may go down a different path.
 8. If using Miro, import the tickets into your project's wiki for documentation purposes, once the backlog is 120% complete! *Keep an eye out for formatting issues.*
-

Tips

- Structure the backlog in more technical way (or in a development way), i.e what would you prefer to develop first and what all features need to be completed before you starting other tickets.
 - Bring all the related features together in an epic, in order to avoid duplication of AC's and estimates. For example, forms behaviour can have multiple custom questions and some requirements may be duplicated across tickets.
 - Separate out common features like Search and Filter, so they can be built as a re-usable component with minor changes (e.g. different filters)
 - When writing Acceptance criteria ensure you are using assertive/definitive language
 - The user should be able to login
 - The user can login
 - Use IDs in your story map to maintain referential integrity across the tools we use.
-

Backlog Components

The Requirements Backlog will contain each of the Epics, Stories and Tasks that you have created to detail the development requirements.

Consistent Fields

- **Issue Type:**

- The issue type that is selected in Jira to represent the type of work that is being completed.
- **Epic** – Used to group multiple Stories and Tasks together under a similar goal/functionality.
- **Story** – The most common type of issue, used to detail a piece of work based on a User Story.
- **Task** – An alternate issue type, used when a piece of work can not be tied to a User Story.
- **Defect** – happens during or after development and issue usually an unknown or obscure cause that may need chasing
- **Epic Link:**
 - The point of this column is to capture the Epic Name per issue. This is left blank for the Epic issue type.
- **Summary:**
 - This is the name or title of the issue, including Epics.
- **Description:**
 - This is where the body of the content is put for all issue types. This will be translated into the description in a Jira issue.

Issue Type – Epic

Stories and Tasks are grouped by Epics in the backlog. The Epic should contain:

- **Summary:**
 - Summarise the purpose of the Epic to provide context to the Stories and Tasks within.
- **Personas:**
 - List any user types that will be a part of this epic
- **Assumptions:**
 - Provide any detail about assumptions that have been made for an epic to be possible to implement. This could be the inclusion of a specific integration, database migration etc.
- **Risks:**
 - List of concerns that may arise such as a dependency on another epic, content from the client or risk in technology needed to be researched

Issue Type – Story

Stories describe functionality that a User will directly interact with. Style the Issue Type column for the Story with light green so that it can easily be identified as a story while scrolling through the backlog.

Stories must include:

- **User Story:**
 - The User Story will describe the interaction including the User Persona, the action the User wants to take and the context for the action.
 - It will follow the format:
 - “As a [User Type] I want to [do this action] so that [action can be completed]”
- **Background:**

- Provide more information and context to the issue.
- **Acceptance Criteria:**
 - Acceptance Criteria is a checklist of specific, actionable items that a developer must tick off before they have completed a ticket.
 - Should strictly be functional requirements, this is due to the nature of non-functional requirements being almost impossible to predict
 - An example of a functional requirement would be:
 - A system must send an email whenever an order is placed
 - A related non-functional requirement for the system may be:
 - Emails should be sent with a latency of no greater than 12 minutes from such an activity.
- **Implementation:**
 - Implementation notes will assist the developer in where to start with the ticket. They will not be prescriptive, but rather general notes and links to resources that might be helpful. It is also important to note when a story or task relies on another story or task being completed first in the implementation notes.
 - List out any risks that you have identified. This could be risks around data segregation, unexplored technical functionality etc.
- **Test plan:**
 - This is where you will leave details on how/what should be tested
- **Blocked By:**
 - This can allow you to block the current ticket by another one, or an epic or the client providing access /information need to start work. This ticket cannot come in an iteration before the thing that is blocking it. you can have it a part of the same iteration but order matters
- **Requires:**
 - This is less likely to be used but occasionally you'll have obviously distinct tickets that may have some level of coupling and would likely be implemented on the same branch, adding the ticket here can just indicate to the developer that the features are paired or grouped and likely should be done together or in the same iteration to make sure they flow well together.

Issue Type - Task

- **Tasks** describe functionality that a Users do not directly interact with. Style the Issue Type column for the task with Light Blue in order to easily identify Tasks while scrolling through the backlog.
- Tasks follow the same format as Stories, with the exception of the User Story which is replaced with a Description. The Description outlines the objective of the task in 1-2 sentences.

Story Map IDs in Scope

It can be challenging keeping track of stories during scope as we work across multiple different

platforms such as Excel, Miro, Confluence and Jira. IDs are assigned automatically once stories are created in Jira but as this is at the end of the process, we have no way to manage stories as they are renamed, reordered, removed etc. To resolve this we can manually add and maintain a set of IDs prefixes on our story maps. These IDs will exist as part of the story name and as such, will be carried across to all the various tools we use.

Video

[Story Map IDs in Scope \(internal link\)](#)

Using IDs

Story IDs refer to stories, the epic prefix on a story ID is only to be used to ensure uniqueness, it should not be used to guarantee the epic.

- In discussion
 - When discussing stories within the team or with the client, a story map ID can be used in place of a name. This can be useful in avoiding ambiguity between features that similar names, as well as reducing the cognitive load required to track or remember issues.
 - Story map IDs persist throughout the life of a story, so while references to “Left sidebar navigation” may become ambiguous as new navigation features are added or changed, references to [3.1] will remain unambiguous.
- Story linking
 - Linking between stories in the Blocked by section can become useless if the name of a story changes. Using IDs allows for the link to remain beneficial even beyond

Creating IDs

1. Starting at the top, move down the list of epics numbering each one starting from 1 following a 1 increment for each new epic (i.e. 1,2,3,4,5, ..., n, where there are n epics).
 - Prefix each epic name with this number, for example, the first one in image above has the epic ID 1, so has been prefixed with [1].
2. For each epic, starting on the left most story, number each story starting at 1 and incrementing by one each time.
 - Prefix each story name with both the epic ID and the story increment. This makes the story ID. The first story in the first epic would have the ID [1.1] which identifies it as such.

Maintaining IDS

The following guides can help with maintaining the integrity of IDs.

Scenario	Action	Comments
Moving a story to a different epic	ID does not change	A story ID identifies the story, not the epic, once an ID has been set it should not be changed as changing it breaks the reference. Additionally, if an epic already has a story increment that is the same, maintaining the old epic increment ensures uniqueness.
Deleting a story	ID does not change	An epic may start on a story ID that is not 1, for example, if we delete the first story in epic [1], we may end up with the epic stories starting from [1.2].
Reordering a story	ID does not change	Story IDs should not be used to identify ordering, the ordering is to ensure uniqueness only.
Exporting stories to another platform (i.e Excel, Jira)	ID remains as part of the name	This ensures that IDs remain useful and relevant throughout the lifecycle of the issue.

Exceptions

As with all things, there exists exceptions to the above rules. If early in the scope, a scoping team member can decide to update IDs after stories have been rearranged/reorganised if and only if they also:

- Search the story map for any references and also update them (this can be done with Miro search)
- Accept that any old references are now completely incorrect (it is encouraged to not do this once IDs have started to be used)



Designtivities

Designtivities are an exciting set of activities that are specifically curated to empower product designers and user experience (UX) designers in software development. They are like a treasure trove of tools and resources that provide designers with everything they need to create amazing designs that truly delight users. These activities are tailored to the unique needs of designers, and include activities such as user research, wireframing, prototyping, and usability testing that allow designers to truly understand their users and design accordingly.

Designtivities are the key to creating consistent, user-centered designs that are both meaningful and functional. They provide a set of established methods and techniques that can be used by designers to streamline the design process and make it more efficient and effective. With Designtivities, product designers and UX designers can take their designs to the next level and create software that users love to use.



Lean UX Canvas

A facilitation tool for Brief meetings, that allows us to create a customer-centric Problem Statement for the upcoming scoping phase.

Activity Details

Activity Lead

Account manager or product designer

Who's typically involved?

Account manager, client, and product designer

No. Participants

2-3

Est. Completion Time

1 hour

Template Link

- Internal reference: Activity is located on the [Canvas miro template](#)
 - External reference: [Template example](#)
-

Steps

1. Using the template, fill in each area on the left during the brief meeting with sticky notes – make sure you get enough information down to form a problem statement and define the top business and user outcomes.
2. After the brief meeting, refine the problem statement and business and user outcomes before passing them over to the Account Manager.

Business outcomes

Some questions you could ask:

- What will indicate that we solved the problem?
- What ways can we quantify the success of this solution?
- What are some things we can track?

User outcomes

Some questions you could ask:

- What benefit would users gain from this solution?
 - What ways can we quantify the benefits?
 - How might this solution optimise their process?
-

Tips

- As soon as the clients start talking, start taking notes in the miro board. This way you already have content on there when the time comes for you to share your screen.
- Aim to have notes in all of the left hand side boxes. The hypothesis is tricky, but important. Try asking “What’s blocking you today?”, “We’ll fix this for you first, and then move onto the next priority after that”.
- Don’t feel the pressure to finish the problem statement in the meeting. You just need to know enough, so that you can form it yourself after the meeting.
- If there is more than one client participant, make this activity more interactive by getting participants to note down their business and user goals on sticky notes. Then discuss each one and decide which are the top three!



5 Love Languages

Since we spend a lot of time with our clients, we need to build a strong connection to understand them and their needs. Using 5 love languages to give the attention that our clients appreciate would help build that relationship.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer and client

No. Participants

N/A

Est. Completion Time

Ongoing

Understanding 5LL for our relationship with clients

With our lines of work, we want to build long-lasting relationship with our client and the 5LL can help us understand them and their needs better. Ensuring that we keep our common goal in mind in everything we do can help us achieve them more efficiently. 5LL allows us to give attention to our clients in the way that they want to receive it as no two clients are exactly alike.

How do you know your client's 5LL?

As much as we would love to, we can't read our clients mind however; we can see how they react to our actions. The best way to know your client's 5LL is to try express them all and see which one caught makes the most impact to the relationship! Here are some of the things you

can do to express 5LL in a work setting.

Quality Time

Ongoing 1:1s

- Interact with your clients often in a casual or relax setting.
- Have an ongoing session that is consistent; the more you interact with them, the easier the conversation will flow.

🌟 In certain projects, we do our backlog refinement session every Monday morning in person. It allows us to have a more relax and open conversation.

Cursor Cuddles

- Work in the same document as your client
- Allows them to give you feedback as you go and have them feeling like they are involved with the process.
- Building alignment that you are in the same step together

🌟 Invite clients to the Figma file to observe the progress and open for them to leave comments.

Session Invites

- Send them calendar invites for scheduled meetings early!

Physical Touch

Physical Immersion: How to reduce the distance between you and your clients

Share Raw Data

- Share raw data with the clients such as videos/recording of the session, notes taken, key discussions
- Clients have a sense of what is going on

Visualisation

- Invite them to do research/testing in the field; help people to visualize the pain points.
- Helps them feels immersed, a lot less convincing for us to do.

🌟 Invite the product owner to observe (or conduct!) the user testing session so they get a

sense of any pain point or success points better.

Acts of Service

Summarize results

- Share results in a way that is easy to digest and in a format that is enjoyable for the clients.

What – So What – Now What

- A framework that makes it easy to share heavy insights and research.
- **What:** What are the raw data and insights found
- **So what:** summarizes them and group them to topics
- **Now What:** put them in context (goals, journeys) – business goals and user

🌟 Use the Miro template for What – So What – Now What activity

Receiving Gifts

Give them relevant links

- Hear clients talk about wanting to learn something (whether its relevant to the project or not), find the link and send it to them.
- Makes them feel like you really listened to what they said and actually responded.

Personalized Docs

- Create documents as the clients prefer to receive them (if they prefer have a really detailed document or having everything summarize, maybe include links of the resources, screen shots of example etc).

🌟 Here is an example of a product owner we worked; they originally created really detailed and formal individual documents for us, after some discussions and difficulties of having to go back and forth between the pages, he decided to collate them and summarize them to a different page which made it easier for us to refine with!

Words of Affirmation

Love Bombs

- A miro kit that helps the team to share their positive affirmation to their co-worker.

- When project gets to a point that it gets tiresome, pull up this quick activity to bring your co-worker spirits up!

Template

- Internal reference: Activity is located on the [Miro template](#)
- External reference: [Template example](#)

Highlight Wins

- Highlights the wins every now and then with your team to give them reassurance that you are moving forward to the ideal solution.



NOTES

The activity kit is inspired by a Presentation from Vindhika Basal; can be accessed [here](#)



Assumptions List

Assumptions can be things that the team believe to be true for how the software solution will perform. As you move throughout scoping, make sure you record and gain clarity over them. Assumptions are never wrong, just a starting point to ensure we are on the right track.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Clients + scoping team

No. Participants

2-5

Est. Completion Time

30 mins

Steps

1. Write out a list of project assumptions as statements using the template below
 2. Use the categories below as guidelines
 - Resources – people, materials, or facilities needed to complete the project
 - Delivery – what’s intended to be delivered
 - Budget – estimated cost of the project
 - Finances – funding to complete the project
 - Scope – the scope of what’s to be delivered
 - Schedule – tasks, durations, and dependencies needed to complete the project
 - Methodology – the approach you’ll take to completing the project
 - Technology – this could cover software development, platforms, environments, and infrastructure
 - Architecture and design – architecture and design approach your team will use
 - Data and performance requirements – number of users expected & amount of data
 3. Address the list in the next scope meeting
 4. Discuss your assumptions and record the response. Add additional notes for clarification
 5. Mark Y/N against assumptions as you go
-

Tips

- Tips from Izzy – Although it is best to clarify assumptions as early as possible during scope, add to this list anytime throughout the scope process... Whenever there is an assumption made by the team that needs confirming or clarifying in your next meeting.
-

Template

Assumption	Y/N?	Notes
Resources – people, materials, or facilities needed to complete the project		Additional notes for clarification
Delivery – what’s intended to be delivered		
Budget – estimated cost of the project		
Finances – funding to complete the project		
Scope – the scope of what’s to be delivered		
Schedule – tasks, durations, and dependencies needed to complete the project		
Methodology – the approach you’ll take to completing the project		
Technology – this could cover software development, platforms, environments, and infrastructure		
Architecture and design – architecture and design approach your team will use		
Data and performance requirements - number of users expected & amount of data		



Breadboarding

A concept borrowed from electrical engineering to help us to design at the right level of abstraction. This activity will help you layout all of the components of UX, without getting caught up on the UI.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Clients + scoping team

No. Participants

2-5

Est. Completion Time

20 mins



Terminology

- **Places:** These are things you can navigate to, like screens, dialogs, or menus that pop up.
- **Affordances:** These are things the user can act on, like buttons and fields. We consider interface copy to be an affordance, too. Reading it is an act that gives the user information for subsequent actions.
- **Connection lines:** These show how the affordances take the user from place to place.

Steps

1. Discuss the problem you are solving and determine where to start, this will be your first place. Write it down and underline it. Eg. Invoice screen.
2. What will users need to do on this Invoice screen? Write these affordances down underneath the underlined place. There can be multiple affordances. Eg. Turn on the auto-pay feature.
3. Where does this 'turn on auto-pay' button take us? The answer will be your next place. Eg. Setup Auto-pay. Write it down beside the first place and then draw a connection line between the affordance and the new place its going to take the user to.
4. Talk about what belongs on that screen now, and continue the pattern until the user journey is complete. But you will notice, that just figuring out what affordances belong under each place will provoke debates and discussions about what to build.
 - You may realise that there are different ideas and opportunities that arise - since we're using such lightweight notation, and we aren't bogged down with wireframes, we can quickly jump around and entertain different possibilities!
5. Once you get to a place where we play through the use case and the flow seems like a fit, we've got the elements we need to move on to start defining the project in more detail and perhaps visually.

Source: "Shape Up" by Basecamp. Read the chapter below

Tips

- Jordie - Read the [chapter](#) from the book to gain a deeper understanding & see an example.



Crazy 8's

The best time to do this is when you're stuck and want to get some ideas flowing. This activity prevents you from overthinking as you are working to a time. The focus is on getting the stuff in our brain out on paper.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Group of designers or clients and the scoping team

No. Participants

3-8

Est. Completion Time

10 mins

Steps

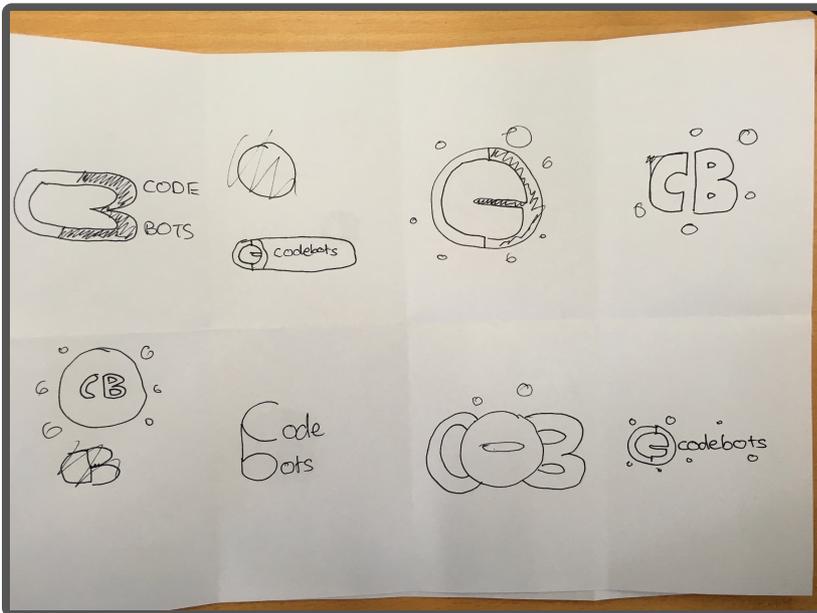
1. (Pre-work) Fold A4 or A3 sheets of paper so you have 8 sections
 2. Give each person a folded sheet
 3. Set a timer for 1 minute per section and ask the group to sketch 1 quick idea
 4. Repeat 8 times
 5. Ask people to present their top 3 or favourite ideas to the group (have someone scribe as they present)
 6. Playback the ideas and get people to vote on their favourite 2
 7. Take the outputs to explore, build on and refine
-

Tips

- Tips from Josephine - Assigning 1 min per sketch helps with the churn. It's about quantity not quality after all, so I've found by the 3rd/4th people tend to loosen up!

Examples

Someone's early ideas for the Codebots logo, completed as part of a Crazy 8s activity.



Codebots Logo Crazy 8s



Custom Components

Custom components are designed according to unique needs, specifications and preferences for a project. This means that the component can't be configured using what's already native or out-of-the-box (OOTB). As such, we want to ensure that during scope we account for and estimate the time to build these custom components.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Clients + scoping team

No. Participants

N/A

Est. Completion Time

N/A

Template Link

On the User Story Map Miro template, Figma Bot Components.



Steps

1. Review OOTB components on the 'Bot Components' template on Figma
2. Decide if you need to create a custom component or update an existing
 - If NEW - create the master component on the 'Custom components' artboard
 - If UPDATING - add a comment (Shortcut 'C') and note what is different
3. Assess the work with the developer
 - Create a task on the USM under the epic 'Custom components'
 - Fill out the details of the task together
4. Prioritise on the epic as you would normally
5. Developer provides estimates on each



Design Principles

As you build out ideas, you'll notice that there are unifying elements that start to guide the design. These are the principles and they can help build a shared understanding of what "good" looks like.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Clients + scoping team

No. Participants

2-5

Est. Completion Time

20 mins



Steps

1. Consider the core principles that underpin the product or service
 - Leverage existing client documents, brand strategies, etc
 2. Frame these as positive statements that might tell you how and what to design
 - E.g. The systems do the heavy lifting – this implies the user shouldn't have to do much manual work
 3. Come up with at least 3-5
 4. Ask the group:
 - Are they short and to the point?
 - Do they describe just one idea?
 - Do they need to be broken down?
 5. Review and ensure they cover the key aspects of your solution
-



Tips

- Tips from Jordie – use these principles to influence the UI and UX you design. For example, if “Build for Accessibility” is a principle, you might consider creating a high contrast mode.



Examples

Here's one we did for a client:

Design principles

Created by Josephine Nguyen
Last updated Tue 12/May/20 - 1 min read - lit 5 people viewed

Design principles are quick, memorable recipes that help keep design and development consistent. They describe the most important elements of your solution and give integrity and form to what you're designing.

1. Systems do the heavy lifting
2. Build for accessibility
3. Speak the same language
4. Enable connection
5. Data at the centre

NGR Design Principles



Discovery Interviews

For uncovering first-hand what the market really needs. It allows you to open a dialogue with users on what's working, what's not working and what the industry needs.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer, product developer and the users

No. Participants

3-5 users

Est. Completion Time

30 mins per session

Steps

1. Set the criteria for who you want to interview (3-5 users is a healthy amount)
2. Recruit some participants. Ask your client if they have users in mind first, otherwise you can get creative into order to find your participant (use the email templates below to make contact with users).
3. Prepare a list of questions (use the template below)
4. Decide on your method (face to face, over zoom or via phone)
5. Don't forget to record your sessions (screen recording, audio recording or have a dedicated scribe)
6. Once all of your sessions are complete, document your key insights

Suggested activity items:

- Motivation Card Sorting
 - User Flows
-

Tips

- Tips from Jordie: When you're talking with users, keep the conversation casual and make them feel comfortable. If they go off script and discuss other topics, its because they're passionate! See where the conversation goes and rein it back in when it feels natural.

Examples

- Check out this article on [Getting ready for user interviews](#)
-

Template

This template has a few questions to get you started, but you will need to customise them for your scenario. Try to keep the interview to 30 mins max.

Objectives

- To identify the pain points and areas of opportunities
- To determine level of interest in potential features
- To ... (provide more objectives as you go)

Interview guide

1. Introductions (5 mins)

- Introduce who we are and what we do
- Set the scene

2. Getting to know you (5 mins)

- Tell me about your role

3. Journey (20 mins)

- Walk me through your day
- Tell me about a time you ...
 - What's the hardest part?
 - Have you been able to make that easier?
- How much time is spent doing ...
- What is the biggest pain point?
- Where do you see opportunities for improvement?

Participants

Schedule	Name	Background	Notes
Time of interview	Name and contact method	What do you know about them?	Interview notes...



EMAILS

Template to send to Client, before reaching out to users

Good afternoon [name],

Thank you for a productive scope session today/yesterday.

We've spent the day improving the user journeys and writing out our list of assumptions. We also worked on preparing a script for the user interviews (below). [Name], would you like to take

a read through the script and let us know if there is anything else that you would like to ask?

[AnotherName] do you mind send us through the contact details for everyone?

- [User] from [Clients client]

Where possible, its best for you to get in contact with each of them first, to let them know I'll be reaching out shortly.

Looking forward to hearing back from you.

Kindest regards,

[MyName/sig]

Template 1: General

Hi [name],

[KeyPerson] has passed on your details and recommended you as a great candidate for our study.

My name is [MyName] and I'm a Product Designer from WorkingMouse 🙌 . As part of my role I get to chat to people like yourself and learn about the things you like or don't like when using software.

I'm working closely with the [ClientName] team to uncover ways we can enhance the users' experience.

What we need from you: You won't need to prepare anything before the session. We're mostly keen on having a chat and hearing about your experience using [tool]!

Dialing into a Zoom call

- Visit [Zoom Link]
- Enter the meeting ID: [000 000 000]
- Launch the Zoom app or simply 'Join from your browser'

Don't hesitate to reach out if you have questions.

Kind regards,

[MyName/signature]

Template 2: General with time slots

Hi [Name],

I'm a Designer at WorkingMouse and we're working with the [ClientName]'s team to build them [ProductName]. [KeyPerson] mentioned you might have some helpful feedback for us.

We'd like to have a discovery interview with you at some point this week if possible? The session will last about 30 minutes, and can be held in person or via Zoom. Our office is located in Milton or we can come to you if you're close by.

To participate, we have the following time slots available.

- Today: Anytime after 12pm
- Thursday 26th November: Anytime after 12pm
- Monday 30th November: Anytime

If you're unable to make it in the next couple of days, we will be performing a second round of interviews at the end of next week if you'd like to join in then.

Looking forward to meeting you

Thanks,

[MyName/signature]

Template 3: Returning clients ready for their next build

Hi [Name], how are you going?

I hope you've been well since the last time we caught up. The [ClientName] guys have been using their [NewTool] this year, and they're ready to start building the next phase: an [NewProduct Tool].

[KeyPerson] mentioned that you and your colleague [Name], would once again be the perfect people for us to talk to for some feedback. I wondered if you'd be free in the next few days for a chat, like we did last time?

The session will last about 30 minutes, and can be held in person or via Zoom. Our office is located in Milton or we can come to you if you're close by? To participate, we have the following time slots available:

- Today: Anytime after 12pm
- Thursday 26th November: Anytime after 12pm
- Monday 30th November: Anytime

If you're unable to make it in the next couple of days, we will be performing a second round of interviews at the end of next week if you'd like to join those.

Once again, we appreciate your time.

Regards,

[MyName/Signature]



Icons

Icons are an important part of your user interface. Some projects only require some basic icons, whereas others call for their own custom set.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer

No. Participants

1

Est. Completion Time

1 hour - 1 day

Steps

1. We predominantly use Material Icons for our projects, given its wide variety and ease of use. So when you start out a project, browse through [Material Symbols and Icons - Google Fonts](#) before you turn to creating your own custom font.
 - If you are happy with using Material Icons - read our Material Icons Tips page.
2. If you can't find what you're looking for in Material Icons, you can create your own. For a full guide on how to create your own, read Custom Icons Tips.
 - Please note that you can use both Material Icons and a custom icon font in your project.

Please note that we previously used [Codebot's Lightning Icons](#) for our projects. We've moved away from this, as Material Icons are easy to use with Figma. If you would like to use them, take a read of the docs.

Tips

- Tips from Jordie - Custom icons will add that extra layer of detail to your project, but please note that it can take 2-3 days to create a basic icon font. Good news is, the font can be made during development.

Examples

- We used Codebot's Lightning Icons on [SuzanneStays](#)
-

Material Icon Tips

If you're planning to use Material Icons in your project, here is all you need to know.

Designers (During scope)

1. We have published this [Material Design Icons Library](#) in our Figma team for you to easily add their icons to your project.
 - To activate the library in your project, go to Assets > Libraries > Toggle on 'Material Design Icons (Community)'.
2. Once activated, you can search for icons by their name and drag them into your project
3. Material Icons come in 5 different styles: Filled, Outlined, Round, Sharp and Two Tone. Access these options from the Component settings panel.
4. The only thing left to do, is ensure you document where your icons came from in the Design System, so that Developers are aware how to implement them.

Developers (During Iteration 0)

1. Review the docs at [Material Icons Guide | Google Fonts | Google Developers](#)
 - It seems that for web applications, the best way to implement is via an Icon Font for the Web.
 - If you're building a mobile app, you may have to install the font locally. In that case, it may be worth asking the designer to sift through which icons should be included to save file size.



Custom Icon Tips

So you've decided to make custom icons? Take a read of this to see what's involved.

Designers (During scope)

1. While the goal would be to have all custom icons created during scope, sometimes you only have time to make a few. That's alright, you can complete the rest of them during development.
2. However you choose to make your icons, there are a couple of basics to creating a cohesive icon set.
 - Decide on outlines vs fill and round vs sharp
 - **Please note that creating your own two-tone icons, comes with a set of complexities!!**
 - We cannot find any free icon font generators that allow custom duotone icons, only paid versions. Font Awesome allows users to create duotone icons as a part of their Pro account, which costs \$99/year. And unfortunately it seems you will need to continue to pay that license every year to continue to have access to these icons. So that is option 1.
 - Option 2 is to implement these icons as SVGs instead, pro's are we will be able to match the appearance, but cons are that they can be tricky to style for the devs and they can also incur a loading time.
3. When you create icons, you might construct them of a number of lines and shapes - which will look a little like this. It's acceptable to leave them in this state while in scope, as time is of the essence. However, they will need to go through a polishing process before publishing them as an icon font.

Designers (During development)

1. Download the *icons-working-file.ai* illustrator file to fit your icons onto the grid. Use the existing icons in the file as an example.
 - Copy and paste all of your icons into the illustrator file.
 - Polish your icons by turning them into individual shapes, which involves expanding strokes and uniting multiple shapes
 - Once you've got a single shape, right click > Make Compound Path. Compound paths are apparently the best way to prepare svgs for icon fonts.

- After you've finished off all icons, rename each artboard to what you'd like the icon to be called.
 - Then export all artboards as SVGs.
2. When you're happy with your exports, go to Fontastic and login with the details in Bitwarden.
 - Follow their instructions on how to create your own font [Create your Icon Font in seconds - 9000 Vector Icons Available - Free Icon Font Generator](#)
 3. Once published, ensure that you document how to find and install your icon font, in the Design System, for developers read.
 - Save a copy of the 'Install Manually' files on Sharepoint for the dev team to access.

Developers (During Iteration 0)

1. Fontastic have a basic 'How to' guide: [Create your Icon Font in seconds](#)
2. There are two ways to install an icon font that we've made through Fontastic.
 - Via web link
 - Or install it manually into your application: Note, this may be the best option for mobile applications. The designer will have downloaded the manual files and saved them in sharepoint for you to access.



Icon fonts vs SVGs

Defining when something should be icon vs an image is tricky. You have to consider the context of use, how frequently they are used, and also how you use svgs (there are things you can do to improve svg usage, eg. inlining them).

When to use an Icon Font

You should most certainly be using an icon font when:

- The icon you have to implement is a single colour (outline or solid).
- The icon is a part of a set, that will be used repeatedly throughout the application (not as a feature illustration).

When to use an Image/SVG

You may need to consider using an image/SVG when: The icons are duo-tone (this does not include emojis, if your project uses emojis, see [HTML Emojis](#)) The icon is more of a feature image / illustration featured at a specific point in your application

How to create SVGs

Consider: [SVG Symbols](#)



Jobs to be Done

We hire products (and software) to do jobs for us. In order to motivate users to buy what we're offering, we need to understand what jobs they would hire this product for. By using this perspective we are able to dissect what it is a person really needs and discover new ways of doing something.

We're adopting [Clayton Christensen's JTBD theory](#) to assist with our understanding.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Client + scoping team

No. Participants

2-5

Est. Completion Time

30 minutes

Steps

1. During the scope kick-off, go to the 'Jobs to be done' board
 2. For each user group, create a large post-it
 3. For each job, create a small post-it and plot them around the user group
 - Record notes on the situation they're in, the motivation they have and what goal they are trying to achieve.
 - E.g. 'I want to like a track'
 4. Following the kick-off, have a team sync up to formalise the post-its
 - Each 'job' will be written to be a feature
 - E.g. 'Like a track' & 'Dislike a track'
 - Mapping this out starts to give you a visual of how sizeable the work is
-

Tips

- Think about what are people going to stop doing once they start using your product?
 - Are the jobs functional/physiological needs, social/psychological needs or emotional needs?
-

Template

User Groups

- Task manager
- ...
- ...

As a [INSERT-USER-ROLE]

Situation (When...)	Motivation (I want to...)	Goal (So I can...)
E.g. I look at my tanks full of fish	E.g. Have oversight on each tank	E.g. Add more if needed



Lean UX Canvas

A facilitation tool for Brief meetings, that allows us to create a customer-centric Problem Statement for the upcoming scoping phase.

Activity Details

Activity Lead

Account manager or product designer

Who's typically involved?

Account manager, client, and product designer

No. Participants

2-3

Est. Completion Time

1 hour

Template Link

- Internal reference: Activity is located on the [Canvas miro template](#)
 - External reference: [Template example](#)
-

Steps

1. Using the template, fill in each area on the left during the brief meeting with sticky notes – make sure you get enough information down to form a problem statement and define the top business and user outcomes.
2. After the brief meeting, refine the problem statement and business and user outcomes before passing them over to the Account Manager.

Business outcomes

Some questions you could ask:

- What will indicate that we solved the problem?
- What ways can we quantify the success of this solution?
- What are some things we can track?

User outcomes

Some questions you could ask:

- What benefit would users gain from this solution?
 - What ways can we quantify the benefits?
 - How might this solution optimise their process?
-

Tips

- As soon as the clients start talking, start taking notes in the miro board. This way you already have content on there when the time comes for you to share your screen.
- Aim to have notes in all of the left hand side boxes. The hypothesis is tricky, but important. Try asking “What’s blocking you today?”, “We’ll fix this for you first, and then move onto the next priority after that”.
- Don’t feel the pressure to finish the problem statement in the meeting. You just need to know enough, so that you can form it yourself after the meeting.
- If there is more than one client participant, make this activity more interactive by getting participants to note down their business and user goals on sticky notes. Then discuss each one and decide which are the top three!



Legacy system demonstration

This activity builds understanding of current processes, pain points and areas of the existing application which are important to users.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Client, product designer, and product developer

No. Participants

3-4

Est. Completion Time

1 hr

Steps

1. Organise for your clients to present a system demonstration, typically in the 1st or 2nd scope meeting.
 - Take notes of the walkthrough and record the session if possible.
 - Ask the person doing the demonstration to walk through the core use cases.
 - Identify problem areas – these are opportunities for you to improve on!
2. Ask the Product Owner if the scoping team can have access to the legacy application during scope so it can be referred back to.
3. It is critical that you gain a deep understanding of the legacy system, as we don't want any core features to be left behind. Sometimes what the Product Owner and the end users perceive as core features, can be different. So it is important that you gain insights from both parties.
 - Organise Discovery interviews sessions with existing users of the system. However, you may want to extend the sessions to include product demonstrations from them too.
 - In cases where the system has A LOT of features, and you're finding it hard to gain clarity over which ones are valuable to users, you can run a [Feature Matrix activity](#), where you ask them to rank existing features by their easy of use and usefulness.

Questions to Ask

- What are the different tasks you perform on the platform?
 - Why do you perform this task?
 - What happens outside of the platform for you to perform this task?
 - What are some of the things which work well for you when you carry out this task?
 - What are some of the things which don't work so well?
-

Tips

- Tips from Izzy – Before commencing scope ask the client to prepare an existing product demonstration to present to the team in the first meeting.
- Tips from Izzy – A legacy system demonstration early on in the scope process helps the team understand what the users are currently working with.
- Tips from Izzy – Start to sketch up an initial user flow of the legacy system on your tablet, notepad or whiteboard while the client is going through the demonstration. Be sure to include what happens outside of the platform to get a full picture of the user flow.



Motivation Card Sorting

This activity can help you gain clarity on what things matter most (and least) to people. It may be used to guide decision making as the product evolves.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer, users/client

No. Participants

2-5

Est. Completion Time

30 minutes

Template Link

- Internal reference: Activity is located on the [Miro template](#)
- External reference: [Template example](#)

Steps

Conducting with a User

1. Share the link to the Miro board with the participant
2. Get them to order the list from most to least important
3. Ask them to explain their ordering
4. Ask if they'd like to add anything that isn't already in the list
5. Thank them and take a screenshot

Conducting with a Client

1. Ask the group to jot down 10 things they think is critical for the system/app
2. Have each person take turns sticking them on the wall
3. Group the similar ones
4. Ask the Product Owner to rate them 1-10 (1 being the highest priority)
5. Engage in discussion and re-shift as necessary
6. Cull 6-10
7. You now have your top 5 priorities

Tips

- Tips Josephine - You can also run this during user testing to see how someone might sort navigation items to build out an IA, you'll just have to rejig the template.



Personas

Personas are created to represent the different user types that might use your product. Creating personas can help you step out of yourself and to recognise that different people have different needs and expectations.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer, product developer, client

No. Participants

3-5

Est. Completion Time

30 - 45 minutes

Steps

1. Identify each persona based on the user groups that have been identified
2. Name each persona
3. Use the template to start to add details about each persona
4. Validate each persona once discovery interviews have been completed
5. Highlight any changes to personas post discovery interviews

For a more comprehensive guide to personas, this is a great article [Personas - A Simple Introduction](#)

Tips

- It is useful to create the personas early on based on client assumptions of target users and then to validate these personas during the interview stage. Generally, we find that the client is right on the money or at least very close to the mark. If things deviate significantly from the initial write-ups, then it indicates that further research on the users is needed on the client side.
 - As an example, your client might insist that their key user groups are office workers who will use the application daily to access training modules. Interviews might reveal the opposite - that it is a completely different group who need support, and that they are not interested in training modules, only time tracking.
 - Continue to develop the personas as you learn more about your users.
-

Template

Persona Name	User Group	Demographic Information	Description	Goals	Benefits of the application	Frequency of Use
Jim Preston	Manager	38 years old. Married. Over 10 years in the industry	Jim Preston is a manager at CyberCode. He wants to ensure his team are always submitting correct reports but is so busy with project management work he can't always chase them up. He needs something to help manage the workload.	To have an easier work day	Simplifies staff management. Staff autonomy means Jim can focus on other task	Monthly



Project Focus

Often, our clients will come to us with a problem statement that could unfold in a variety of ways. Because only a finite amount of functionality can be scoped in a given build, it is helpful to provide a project focus document.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer, client, and product developer

No. Participants

3-5

Est. Completion Time

30 minutes

Steps

1. Using the attached template or on a whiteboard, list out project focus areas that have been discussed or raised by the client
 2. Add a description of each focus area – how the users would interact with it and why it is relevant to the organisation
 3. List out the benefits of this focus area for the client’s business and its users
 4. List out any risks associated with the focus area – this could be technical concerns, competitors or marketplace doubts
 5. If necessary, add any additional notes or discussion points that come up for each focus area
-

Tips

- Perform this activity only when you have gauged that the client’s expectations exceed what is realistically possible in a single scope
 - Take the time to walk the client through each possibility but also indicate which focus you would recommend
 - Ensure that the developer participates during both the write-up and presentation of this document
-

Template

The Project Focus document is a table which describes the various possible combinations of functionality and their risks and benefits. The client should select one of the listed focuses so that scoping can get back on track. As can probably be imagined, this activity should not be carried out with every client but is highly beneficial for clients who have ambitious plans for their application. It not only manages expectations but provides a handy roadmap for long-term development that can be referred to again and again.

Project Focus	Description	Benefits	Risks/Considerations	Additional Notes
(Give this focus a name...)	A description of the focus – the functionality it involves, how users would interact with it and why it is relevant to the client organisation.	List of benefits vis a vis the client/users	Risks associated with this focus, such as technical concerns, competitors or marketplace doubts	Any additional notes can be placed here.



Red Routes

Use this activity to assist with prioritisation of user stories. By sorting out stories into their use cases we can better determine what we need to build first and what will give the most value to our first group of users.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer, product developer, client, account manager (optional)

No. Participants

3-5

Est. Completion Time

60 minutes

Template Link

- Internal reference: Activity is located on the [Miro template](#)
 - External reference: [Template example](#)
-



Steps

1. Use the Miro Board template or redraw it on a whiteboard.
 2. Invite all members of the team together for the activity.
 3. Create a sticky note for each feature that you wish to prioritise.
 4. Go through feature and determine how often it is used and by how many of the users, then add the feature to the map in the correct cell.
 5. Discuss why the story or feature is added to that cell and justify your decisions with research if needed.
 6. Once all the stories are laid out, focus on the stories in the red cells and you can use this to determine what the project needs to focus on first.
-



Tips

- Tips from Izzy - Get your clients to start thinking from the perspective of what the customer wants to do rather than what they want the customer to do.
- Tips from Izzy - You may want to colour code the sticky notes by user group.



Screen Resolution

Determine which devices our users will be accessing the platform from. This activity provides clarity over which screen size the platform needs to be optimised for first. It should be completed in the observe phase of scope after the discovery interviews.

Activity Details

Activity Lead

Squad lead

Who's typically involved?

Squad lead + scoping team

No. Participants

2-4

Est. Completion Time

30 minutes

Steps

1. Define each User Group
2. During Discovery interviews, find out what devices the users will be primarily using to access the platform
3. Record the results in the template provided. Highlight the primary screen resolution/s that we need to optimise the platform for first
4. Create a ticket in the backlog as a task
5. Estimate the time required for responsive styling
6. At the end of scope, use this information to complete the scope to development handover checklist

Extra Resources

- Common Breakpoints: [What media query breakpoints should I use?](#)
- How to calculate the right measurements for design across devices: [Device Metrics for Any Screen - Material Design](#)

Tips

- Tips from Izzy - Once you have filled out the table, be sure to discuss which screen sizes that are most important to design for first, with your client. Highlight these in your table.
- Tips from Izzy - Consider each screen resolution size that is required. You may need to create a ticket for each screen size, especially if the screens are vastly different.

Template

User Group	Device	Screen Resolution	Frequency of Screen Use

Common Devices

Device	Width
1080p displays	1920px
13" MacBook Pro (1.5x scaling)	1680px
13" MacBook Pro (2x scaling)	1440px
HD laptops (768p)	1366px
iPad Air, iPad Mini, iPad Pro 9" (Landscape)	1024px
iPad Air, iPad Mini, iPad Pro 9" (Portrait)	768px
iPhone X (Landscape)	812px
iPhone 6, 7, 8 (Landscape)	667px
iPhone 6, 7, 8, X (Portrait)	375px

Common Breakpoints (Mobile First)

Breakpoint	Purpose
(default)	Mobile-portrait
min-width: 480px	Mobile-landscape (and larger)
min-width: 768px	Tablet-portrait (and larger)
min-width: 992px	Tablet-landscape (and larger)
min-width: 1200px	Laptops (and larger)



Trade-off sliders

Formalise a list of priorities to guide decision making as the product evolves. This activity aims to promote a healthy discussion on what is really important and gain alignment within the team.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer, product developer, and client

No. Participants

2-5

Est. Completion Time

30 minutes

Template Link

- Internal reference: Activity is located on the [Scope Kick-off Workshop miro template](#)
 - External reference: [Template example](#)
-



Steps

1. Explain that: Trade-offs are an exchange in which you give up one thing in order to get something else you also desire
 2. Review the categories on the board
 3. Add up to 5
 4. Ask the product owner to drag the dots onto the board. The constraint being that each column must only have one dot
 5. You now have your top 5 priorities - these are useful to keep in mind throughout the project, it makes it easier to manage and meet customer expectations.
-



Tips

- Tips from Josephine - Refer to these when prioritising the story map and determining 'musts' from 'shoulds' or 'coulds'



User Flows

Fundamental for visualising the project and current workflow. Highlight the pain points as they walk you through the journey and call-out areas of opportunity.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Clients + scoping team

No. Participants

2-8

Est. Completion Time

1 hour - ongoing

Template Link

- Internal reference: Activity is located on the [miro template](#)
 - External reference: [Template example](#)
-

Steps

1. Ask the client to walk you through the process or demo the system
 2. Get them to:
 - Talk through the people involved, key moments, actions and thoughts
 - Use images or drawings to help illustrate the journey
 - Call out pain points
 3. You will end up with a timeline or a flow diagram
 4. Use this to create a proposed flow that:
 - Highlights exactly how we are going to improve the problem
-

Tips

- It is helpful to think of a User Flow diagram as a visualisation of various narrative branches.
- Create user flows for each user group and update regularly - don't let it get stagnant throughout the scope.
- Walk the clients through the flow thoroughly, being sure to explain each step. It is vital that the client understands that the user flow will inform the prototype - a source of truth during development, second only to backlog.



User Story Map

This is a visual exercise for defining the work required to deliver the application or system. This map will then be used as a tool to prioritise work.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer, product developer, and client

No. Participants

3-5

Est. Completion Time

1 hours - Ongoing

Template link

- Internal reference: Activity is located on the [miro template](#)
 - External reference: [Template example](#)
-

Steps

1. Break the platform down into epics. These are the different pieces of functionality that make up the platform and is a way to group the user stories.
2. Each piece of functionality is then broken down into user stories or small tickets that can be built independently.
3. Add any additional details or AC's to user stories as you go.
4. Once the User Story Map is fleshed out, walk through each user story with the client, and prioritise each ticket:
 - Pick a story
 - Ask yourself: What value does it provide? Does it align with the intent?
 - Mark the story as a 'Must have', 'Should have' or 'Could have'.
 - Repeat
5. Use the User Story Mapping Board to start building out your backlog.

Keep your project in check

It's easy to get carried away with user story mapping, but if you're trying to hit a certain project size, keep these guides in mind to stay within your timeline:

1. For **POC** (2 weeks dev)
 - 15-20 stories MAX
 - Approx 8 prototype screens
2. For **MVP** (4 weeks dev)
 - 30-35 stories MAX
 - Approx 16 prototype screens
3. For **Product Dev** (12 weeks dev)
 - 85-95 stories MAX

These guides have been based on the theory that approx. 62% of your stories are S(1-4h), 22% M(5h-1d), 11% L (1d-2d) and 2% XL (>2d).

Tips

- Whenever a new feature or functionality is brought up in discussion, add it to the User Story map
- If the Product Success designer is leading discussion, then the developer needs to be curating the board in the meantime - or vice versa.
- Suggest that the map inform the backlog - therefore, the developer assigned to the scope must review the map prior to commencing the backlog artefact
- Use the completed User Flows and Jobs to be done activities to assist in breaking down each piece of required functionality.



User Testing

User testing sessions are critical in understanding how your users interact with your product. Testing insights are recorded once prototype testing sessions have been conducted.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer, product developer, and the user

No. Participants

3-5

Est. Completion Time

30 - 40 minutes

Steps

1. Organise user testing session time with a user.
2. List out a set of tasks that the user needs to complete in the prototype.
3. Introduce the prototype to the user – explain any limitations of the prototype in its current form. This is not a completed product by any means!
4. Explain the purpose of this session.
5. Ask the user to complete each task, asking questions along the way about different features or their workflow.
6. If they are having trouble completing a task, ask them how they would expect to perform each task. Get the user to highlight any concerns or points of confusion. Take note of these.
7. Embrace silence. Don't be afraid of awkward silences. Silence gives the user an opportunity to ask questions or express their opinion. We want the user to be doing most of the talking!
8. Record the testing session and take note of key insights, quotes, or opportunities for improvement along the way.
9. Once the testing sessions are complete record key insights in the attached template.
10. Present key insights to the client in your next scope meeting.
11. Adjust the prototype as needed.

Email Template

Good Afternoon [Name],

Thanks for your time last week. Your insights were so helpful in helping me understand how I can design a platform assists with your workflow.

We should have a click through prototype to show you later this week which we would love to get your feedback on. Are you available [Day and time]?

We would need about 30-40 minutes of your time.

Thanks again for all your help so far.

Kind regards,

[Your name]

Tips

- User testing should be carried out once wireframes/prototype are in a workable state
- You can supply users with a list of tasks to complete or use an external service to set up the testing environment
- Consider testing not only users supplied by the client but random participants matching the demographic.
- Testing Sessions could be done in person, testing platforms or online via a program like

zoom. If you conduct the interviews online, ensure that you send the users the link to the prototype and get them to share their screen.

Template

Purpose

Write a bit about the objective of the research study. What did you set out to achieve? E.g. To test the usability of the proposed design and understand how simple the process flows are to pick up.

Recruitment

Write a bit about how you recruited the participants and what the criteria was in order for them to participate in the study. E.g. We were interested in testing with users who:

- Have recently been onboarded
- Mix of technical and support operators
- Service multiple types of requests

References

- [Link to recordings / raw notes](#)

Notes

Key insight	Observations	Quotes	Opportunities
E.g. #1 Participants found certain terminology confusing	P2 didn't quite understand what the info message was asking them to do P3 expressed the major learning curve during onboarding P3 looked frustrated when they couldn't figure out how to progress	"I'm not to sure how that word relates in this context but I see it a lot" (P2) "It's difficult for newcomers to pick up on this learned terminology when it doesn't make sense" (P3)	Simplify the terminology. E.g. changing 'add entity' to 'add user record' Create a terminology table for internal team to assess
#2 Participants loved the real-time updates on the dashboard	What did you see	Quotes from the individual	What can be actioned



Warm-ups

These are some activities you can use as a warm-up, ice breaker or just to get the mood buzzing. After the exercise, take some time with your team to reflect on the purpose/takeaway message.

Activity Details

Activity Lead

Product designer

Who's typically involved?

Product designer, product developer, account manager, and client

No. Participants

2-5

Est. Completion Time

15 mins

Finding Connections

Purpose: This exercise involves bringing together ideas that serve very different needs or interests to form a new concept. Its going to get us into a creative thinking mindset and encourage out-of-the-box ideas.

Number of participants: 2+

Duration: 10 minutes

Materials: A way to show images to the group, potentially use a Miro board OR bring physical objects to the meeting instead of using photos!

Steps:

1. Pre-meeting: Collate 6-10 seemingly unrelated photos of objects, and organise them into a moodboard style layout.
2. During meeting: Show the images to the group.
3. Give them 2 minutes, to pick two or more items and explore different ways they can be connected. This is quiet thinking time.
4. At the end of the timer, go around the group and take it in turns to present the items you chose, and how you combined them to make a new idea/product/use!

Rock Paper Scissors

Purpose: To promote positive relationships and team building. Everyone's a winner!

Number of participants: 6+ (you'll want to make sure everyone has a partner)

Duration: 5 minutes

Materials: N/A

Steps:

1. Find a partner for Rock Paper Scissors
2. Tournament best out of 3
3. The losing player becomes the cheering squad for the winner, walking behind chanting their name. The winner seeks out their next opponent
4. Every time a player wins, the losing player is added to their cheer squad
5. The final two players go head to head until one emerges as the champion
6. Everyone cheers and chants their name

Yes, and...

Purpose: To get in the mindset that things take time to build, just like a good story. It especially helps when people build on each others ideas to take them there.

Number of participants: 3+

Duration: 5 minutes

Materials: N/A

Steps:

1. Someone starts off with a statement.
2. The person to the left says "Yes and..." and adds a line to the story
3. Continue for ~2 more rounds (depending on how many people there are and if how the story is going)

Example:

- Player 1: Yesterday I went to Coles...
- Player 2: Yes and I found a bag of giant udon noodles...

- Player 3: Yes and I found 1L of kewpie mayo...
- Player 1: Yes and I saw this recipe that had both of these ingredients...
- Player 2: Yes and I'm hosting a 10 person dinner party tonight...
- Player 3: Yes and I forgot they are all bringing +1s...
- Player 1: Yes and I suppose I will need more mayo...
- Player 2: Yes and there's no more mayo left...
- Player 3: Yes and it's a good thing Woolworths is open.
- FIN

Draw a vase

Purpose: To demonstrate how reframing a question can result in different outcomes

Number of participants: 1+

Duration: 1-5 minutes

Materials: Paper, drawing mechanism (Choose your fighter: pencil or pen)

Steps:

1. Instruct participants to draw a vase
2. Instruct participants design a way for people to enjoy fresh flowers in their homes
3. Ask participants to compare the results

I'm a survivor

Purpose: To encourage creative thinking and team bonding.

Number of participants: >2 teams

Duration: 10 minutes

Materials: N/A

Steps:

1. Split into teams - try to have an even spread
2. Each team decides which 3 things they would take to a desert island for a month [5 mins]
3. Teams come together to pitch their ideas
4. Vote which team is most likely to survive

Balderdash

Purpose: To get the creative juices going and get to know each other. It's a game where you read out a random word and everyone writes a plausible definition for it. You then read out the responses anonymously and everyone votes for which they think is the real one. The responses often tells you a lot about the individual and how they may perceive the world.

Number of participants: 4+

Duration: ~15 minutes

Materials: Pen, paper, vibes

Steps:

1. Pre-meeting: You are the 'Dasher'. Head over to <https://flanneljesus.github.io/balderdash/> and choose a word
 - Use the [Miro template \(internal link\)](#) to setup your word.
 - The QR code will take users to the Menti that you've setup for them to submit their answers (login via bitwarden).
2. During meeting: Inform your players what the word is, by showing them the miro board. Ask them to scan the QR codes with their mobiles.
3. Players write a definition for what they think the word means (2 mins max) - it can be funny or serious
4. Players submit answers when finished, and you, the dasher can see their responses come through on Menti
5. Dasher reads out, in no particular order, all responses including the real one
6. Players then blind vote which they think is the real one
7. Dasher announces who voted correctly and who voted for someone else



Processtivities

Processtivities - the ultimate game-changer for your software development process. These activities are specially designed activities are tailored to your specific needs, whether it's Agile, Scrum, Kanban or any other methodology. Say goodbye to the one-size-fits-all approach with traditional activity kits and hello to a customized set of tools and resources that will take your development process to the next level.

Imagine having a clear roadmap for success with activities such as sprint planning, daily stand-ups, backlog grooming, retrospective meetings, and more, specifically designed for your chosen methodology. Processtivities will streamline your process, ensuring efficiency, effectiveness and adherence to the principles of your chosen methodology.

No more guessing or winging it, Processtivities provide a set of established methods and techniques that will empower your development team to reach new heights. Step up your game and take control of your development process with Processtivities.



Async huddles

Activity Details

Activity Lead

Squad/team lead

Who's typically involved?

Entire squad

No. Participants

2-10

Est. Completion Time

10-15 minutes

Steps

1. At the beginning of each work day, each team member compiles a message that details key things they worked on yesterday, will work on today, and any outstanding challenges they are experiencing. (This should follow the template detailed at the top of this document)
2. This message then gets posted to the team channel in the place of a huddle ensuring that a prefix of #huddle exists.

The daily message is a general summary to keep the team in sync and call out for any assistance where needed. Async huddles can be combined with regular huddles or with a reduction of regular huddles. For example, daily messages and in person/remote meetings on Monday, Wednesday and Friday.

Tips

- Prefixing your daily message with the #huddle tag allows for the messages to be filtered and sorted and as such more easily consumed. For example, a Mattermost search term for all Bot-Squad huddle messages would be #huddle in:internal-botsquad.
- These messages are summaries, no need to include every little detail. Try to focus on the top three things for each section.
- Tag team members that you require assistance from to ensure that they are alerted.

Examples

Good morning all, I am WFH today.

Yesterday

- Leads meeting
- Assist with GitLab pipelines
- screenshots for and demo app for

Today

- Document Pipelines
- Review apps and staging
- Review forum and notes

Challenges

- Accommodating multiple time zones with the team huddle has been disruptive, looking to explore alternatives
-



Template

Good morning all, I am [working from home|working from work] today.

Yesterday

- Note

Today

- Note

Challenges

- Note



Retrospective

After completing a milestone such as a scope or development iteration, its important to reflect on how it went. That way our processes continue to improve.

Activity Details

Activity Lead

Team lead

Who's typically involved?

Internal team

No. Participants

3-5

Est. Completion Time

1 hour

Template Link

[External Miro Board Template](#)



Steps

1. Prepare the whiteboard + sticky notes, or the Miro board template
 2. Set the scene for your team
 - Define exactly what you want the team to reflect on, eg. the scope.
 - Assure all team members that is a safe place to be honest and share their experiences.
 - Likewise, this is not a space for attack people. Retros are constructive.
 3. Explain each of the questions are, make sure everyone is on the same page.
 - What went well? - Things that we should continue doing
 - What didn't go well? - What should the team improve
 - What still confuses us? - What should the team seek clarity on
 4. Give everyone 10mins to complete as many stickies as they like.
 - One topic per sticky
 - Each sticky relates to a 'What went well', 'What didn't go well' and 'what still confuses us' question.
 5. Once the timer is complete, start with the first person and ask them to talk through their stickies for 'What went well'. As they talk to each one, ask them to move it into the correct column on the board.
 6. Continue this for each person in the retro. Then continue this for 'What didn't go well', 'What still confuses us'.
 7. Once the board is complete, ask each member to pick off ONE or TWO stickies that are the most important thing to them that gets actioned.
 8. Use these to determine what your action items are coming out of the retro.
-



Tips

- Jordie says: As the moderator, you should be finding patterns across everyone's stickies, and grouping them together as topics on the board.
- Play some background music during the 10mins writing time if you like!



Roadmap

Similar to estimations, roadmaps offer a rough indication of the project timeline and paint the big picture. With this artefact, clients recognise the direction of the project and what milestones are necessary to get there.

Activity Details

Activity Lead

Squad lead

Who's typically involved?

Scoping team internally, then present to client

No. Participants

2-5

Est. Completion Time

30 mins

Pre-work

- To begin communicating a roadmap, reflect on the scope/experience you just went through
 -
 - Remaining features to scope/build
 - How many features are in the next build?
 - No. of expected scopes left
 - No. of expected builds left
- The length of the scope you've just completed
 - Was 4 weeks enough for the amount of work produced? Do you need less/same/more time for the next scope?
- The no. of stories that came out of this scope

- Will the next scope have less/similar/more?
 - The estimated time for building the scoped work
 - Will the next build be less/similar/more?
 - Key project dates
 - If there is a tight timeframe, there is an option to increase velocity by growing the development pod, if the client wants to.
 - Create additional rows depending on the roadmap options. E.g. demonstrating accelerated development with 2 vs 4 devs. Pair this with a set of estimations and pricing that reflects that.
-

Steps for Miro Timeline

- [Internal roadmap link](#)
- [External roadmap link](#)

Tips

- Tips from Josephine - call out pending activities in the grey box on the roadmap. e.g. external branding.
-

Steps for Excel Timeline

These details will assist you in predicting the flow of the project timeline. To get started:

1. Create a copy of the Estimations roadmap spreadsheet on SharePoint
2. Use the Estimations timeline tab for documenting the roadmap
 - Add rows for each option
3. Use the Estimations Pricing tab for documenting project costs

Tips

- Tips from Josephine - add buffers for pending external activities, such as branding