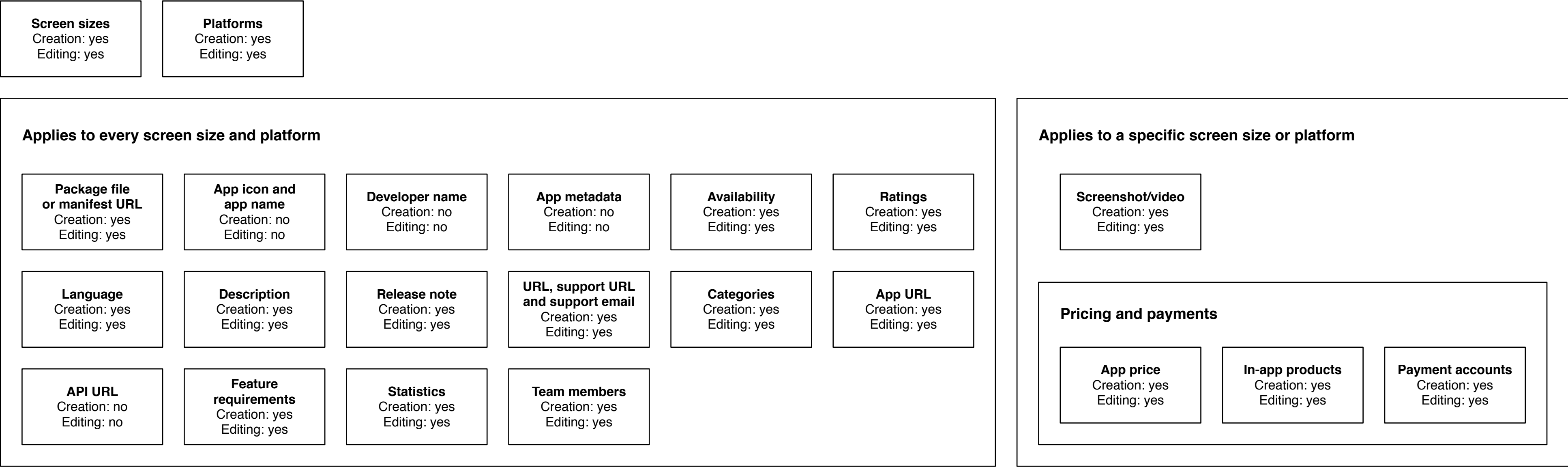


<div>Package file and manifest URL</div> <div><div><div>Package file or manifest URL Creation: yes Editing: yes</div><div>App icon and app name Creation: no Editing: no</div><div>Developer name Creation: no Editing: no</div><div>App metadata Creation: no Editing: no</div></div><div><p>Creating means creating new app Editing means creating new version</p><p>Our goals are:</p><ul style="list-style-type: none">• Pass validation without error• User understands that our primary platform doesn't support Flash<p>We need:</p><ul style="list-style-type: none">• Easy access to testmanifest.com• Easy access to package app equivalent• Flash incompatibility<p>We should consider:</p><p>Eliminating Flash altogether. That is, no app that's created can require Flash. After all, if it requires Flash, it can only run on desktop.</p></div></div>				<div>Screen sizes and platforms</div> <div><div><div>Screen sizes Creation: yes Editing: yes</div><div>Platforms Creation: yes Editing: yes</div></div><div><p>Our goals are:</p><p>Encourage responsive design</p><p>We need:</p><p>Display all three screen sizes with clear terminology</p><p>We should consider:</p><p>Having something encouraging that appears when all three screen sizes are selected.</p></div></div>		<div>Things consumer sees on the app listing page</div> <div><div><div>App icon and app name Creation: no Editing: no</div><div>Developer name Creation: no Editing: no</div><div>Price</div><div>Availability Creation: yes Editing: yes</div><div>Screen sizes Creation: yes Editing: yes</div><div>Ratings Creation: yes Editing: yes</div><div>App metadata Creation: no Editing: no</div></div><div><p>Our goals are:</p><ul style="list-style-type: none">• User finishes adding information in at least one language• User sees the description's limitations (e.g. we will only show the first 8 lines, how many characters per line on each screen size, etc.) and be able to adjust accordingly.<p>We need:</p><ul style="list-style-type: none">• A language selector• An easy way to switch between multiple languages<p>We should consider:</p><ul style="list-style-type: none">• Dropdown menu• Sidebar language menu</div><div><p>Our goals are:</p><ul style="list-style-type: none">• User understands that screenshots are specific to a screen size• User uploads a minimum of 3 screenshots for every screen size• User uploads the correct size for each screen size<p>We need:</p><p>File upload dialog</p><p>We should consider:</p><ul style="list-style-type: none">• A screen size switcher• A smart detector that will detect a screenshot's device, and/or help user assign screenshot to the correct device once it's been uploaded• Linking to MDN resource on screenshot sizes• If this is a test submission, have some sample images at the ready• If we're doing some sort of a preview area, it needs to be pixel accurate</div><div><p>Our goals are:</p><ul style="list-style-type: none">• User understands that description will propagate to every screen size• User sees the description's limitations (e.g. we will only show the first 8 lines, how many characters per line on each screen size, etc.) and be able to adjust accordingly.<p>We need:</p><p>Text input field</p><p>We should consider:</p><ul style="list-style-type: none">• One field to rule them all• Unfortunately, the end result is displayed not as link texts, but as buttons with generic labels.</div><div><p>Our goals are:</p><ul style="list-style-type: none">• User understands that release note is optional if it's a new app• User understands that release note is mandatory if it's a new version<p>We need:</p><p>Text input field</p><p>We should consider:</p><ul style="list-style-type: none">• Hiding release note during app creation• Showing release note only during app editing</div><div><p>Our goals are:</p><ul style="list-style-type: none">• User understands that these fields will propagate to every screen size<p>We need:</p><p>Text input field</p><p>We should consider:</p><ul style="list-style-type: none">• One field to rule them all• Unfortunately, the end result is displayed not as link texts, but as buttons with generic labels.</div></div>						<div>Things consumer don't see on the app listing page</div> <div><div><div>Package file or manifest URL Creation: yes Editing: yes</div><div>Categories Creation: yes Editing: yes</div><div>App URL Creation: yes Editing: yes</div><div>API URL Creation: no Editing: no</div><div>Feature requirements Creation: yes Editing: yes</div><div>Statistics Creation: yes Editing: yes</div><div>Team members Creation: yes Editing: yes</div></div><div><p>Our goals are:</p><ul style="list-style-type: none">• To be able to select one or two• User understands how it can impact listing, although indirectly<p>We need:</p><p>A category selector</p><p>We should consider:</p><ul style="list-style-type: none">• Showing apps in the Marketplace that belongs on the same category as the one that's been selected. It's like doing competitive research• Showing a very visual category selector• Showing explanation for each category• Tying category to ratings. Your business app is probably not going to get an "M" rating. In fact, most apps other than games will need to worry more about user interaction rather than age rating.</div><div><p>Our goals are:</p><ul style="list-style-type: none">• User understands that every app will be given a listing URL• User understands that URL is only viewable by himself until the app is approved• User understands that URL can be listed publicly, or be kept private to a selected audience<p>We need:</p><p>Text input field</p><p>We should consider:</p><p>Having the private/public control be located nearby this URL field</p></div><div><p>Our goals are:</p><p>Unclear. Find out about this from developers.</p><p>We need:</p><p>To just display this information</p><p>We should consider:</p><p>Unclear. Find out about this from developers.</p></div><div><p>Our goals are:</p><p>User understands that these fields will hide their apps to certain set of users with incompatible devices</p><p>We need:</p><p>Feature selectors</p><p>We should consider:</p><ul style="list-style-type: none">• Because there are a lot of feature, consider referring user to MDN for more information• A phone-to-feature matrix that shows the features that can still be checked while maintaining compatibility with as many phones as possible, or as many phones in a specific market as possible</div><div><p>Our goals are:</p><ul style="list-style-type: none">• Encourage user to share statistics to consumer• User understands that statistics are optional• Need more conversation. Do we want to encourage this behavior?<p>We need:</p><p>Checkbox that toggles on/off</p><p>We should consider:</p><ul style="list-style-type: none">• Showing the kind of statistics that is possible to show in a visual way• Limiting the statistics that consumer can see. Maybe developer doesn't want to show absolutely everything, but in the spirit of openness, still wants to show something</div><div><p>Our goals are:</p><p>User understands that team members are optional, but they're helpful</p><p>We need:</p><p>Text input fields</p><p>We should consider:</p><ul style="list-style-type: none">• Hiding team members during app creation• Showing team members only during app editing</div></div>				<div>Availability</div> <div><div>Availability Creation: yes Editing: yes</div><p>Our goals are:</p><ul style="list-style-type: none">• User understands that availability impacts language• User understands that availability impacts price• User understands the trade-off between availability and payment account. The more available = the more payment account forms that need to be filled out later• Encourage user to make their app available in as many regions as possible• Make it possible to make app exclusive to certain regions, if need be<p>We need:</p><p>Country selector</p><p>We should consider:</p><ul style="list-style-type: none">• Grouping by regions</div>	<div>Pricing and payments</div> <div><div>App price Creation: yes Editing: yes</div><div>In-app products Creation: yes Editing: yes</div><div>Payment accounts Creation: yes Editing: yes</div></div> <div><p>Our goals are:</p><ul style="list-style-type: none">• Complete selection of app price for as many regions as possible, as quickly and simply as possible• User understands the trade-off between price granularity and time. The more region-specific their price is, the more time they will have to spend in the form<p>We need:</p><p>Dropdown menu or sliders</p><p>We should consider:</p><ul style="list-style-type: none">• As few price tiers as possible, in order to make selection quick and easy• Grouping tiers or countries together may be a way to reduce the number of tiers• If user wants to customize price in a detailed way, it should be possible</div> <div><p>Our goals are:</p><ul style="list-style-type: none">• If there are only a few in-app products, encourage user to fill them in now• If there are many in-app products, encourage user to fill them in later (during editing) so the creation process finishes sooner and user gets the impression that it's fast• Alternatively, encourage user to fill them in <i>en masse</i>, rather than one at a time<p>We need:</p><p>Text input fields</p><p>We should consider:</p><ul style="list-style-type: none">• Uploading some sort of a text file that contains in-app product names and price tiers, that will be translated in the system (CSV, for example). This is to allow inputting many products all at once.• A way to quickly answer this question: "Would this item I just entered work when actually purchased in the real world?". It would be great to have a way to quickly test and verify a product's validity.</div> <div><p>Our goals are:</p><ul style="list-style-type: none">• Complete sign up for payment account without error, for every possible account• User receives money at the designated time, without error<p>We need:</p><ul style="list-style-type: none">• Lots of text input fields• Payment vendor must be able to verify the correctness of the submitted info<p>We should consider:</p><ul style="list-style-type: none">• Account signup portability. After user sign up for one payment account, he can apply for other payment accounts without filling the forms over and over again.• Just because we store user's payment information, doesn't mean that it has to be insecure. For example, we can fill every billing address and bank info field, but still ask user to enter his bank account number every single time.</div>
---	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--

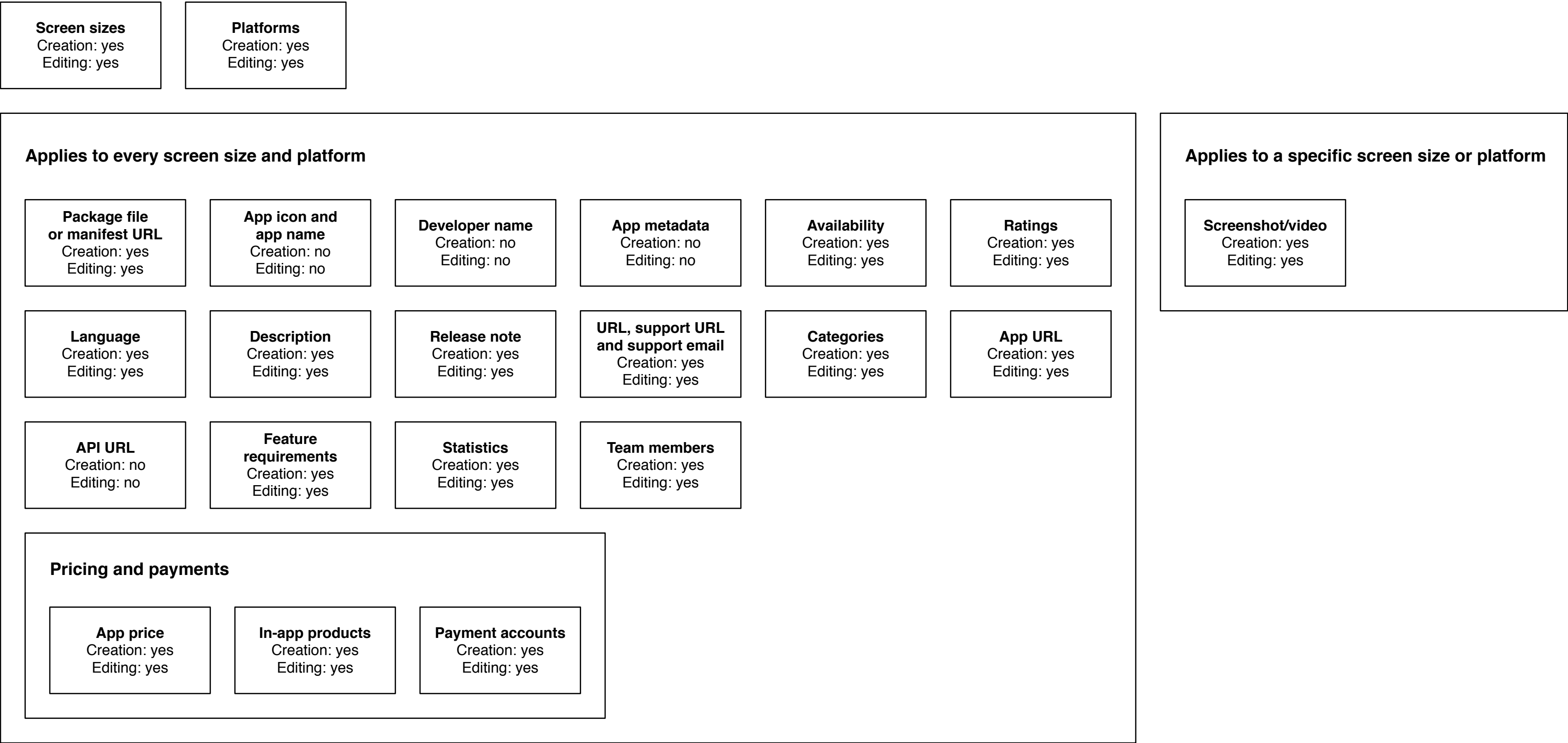
Divided by screen size and platform



Conclusion

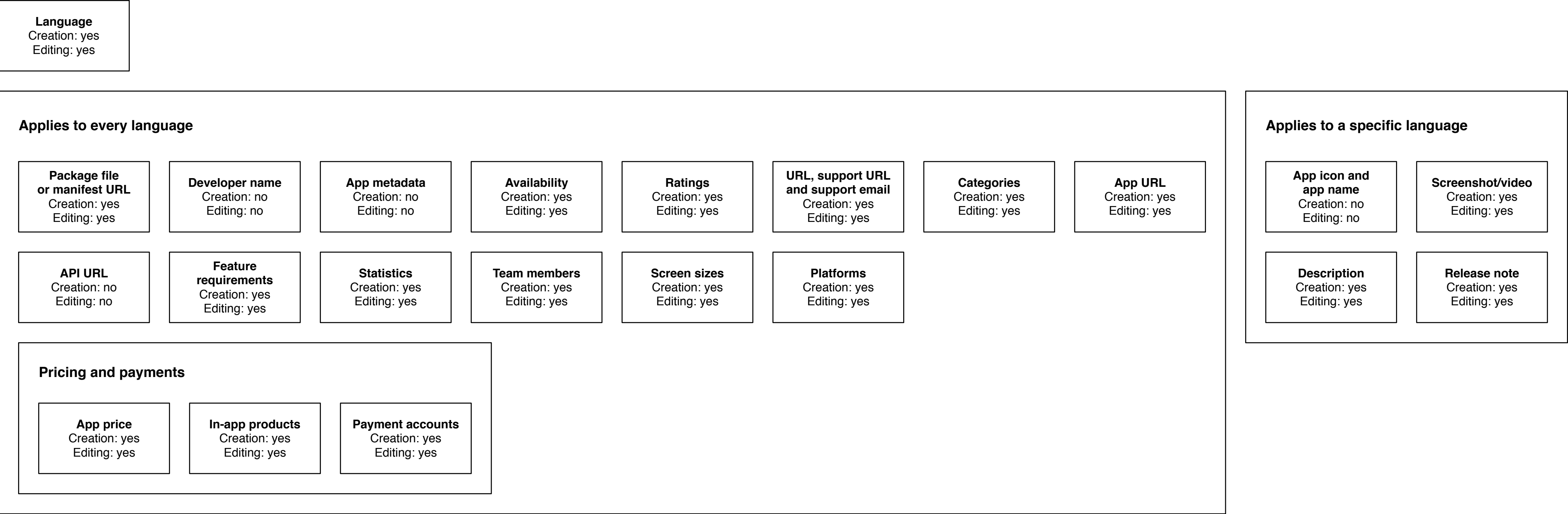
- Screenshot/video is the only visual element that’s screen size-specific.
- This means that, for the most part, our app creation and editing systems can be a system that doesn’t care about screen size, until it’s time to fill out screenshot/video.
- All of pricing is platform-specific. Some countries that support Firefox OS may have a payment provider/price tier that’s incompatible with the payment provider we use for Desktop and Android.

Ideal scenario: pricing and payments apply to every platform



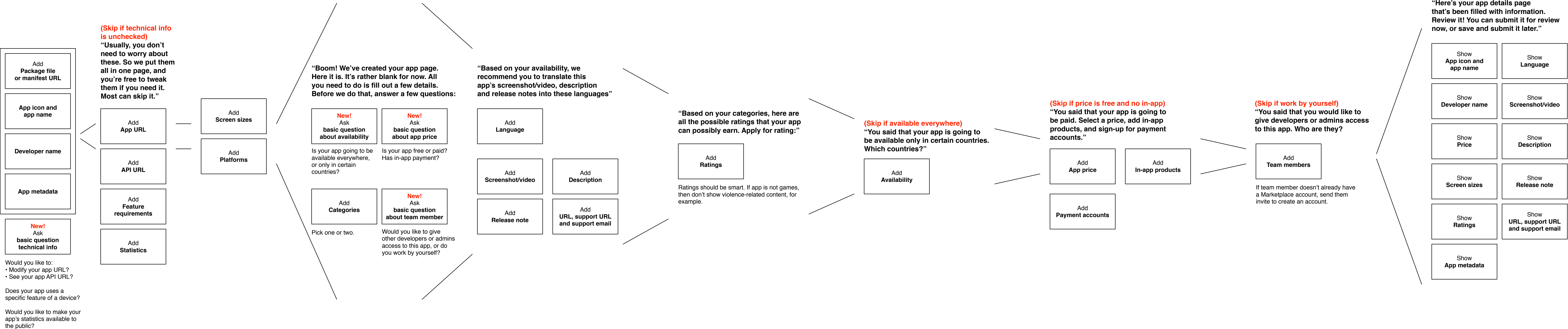
- One payment provider can be used across platforms
- Every platform supports in-app payment
- Every platform supports a set of compatible price tiers that can be controlled from one master switch

Divided by language



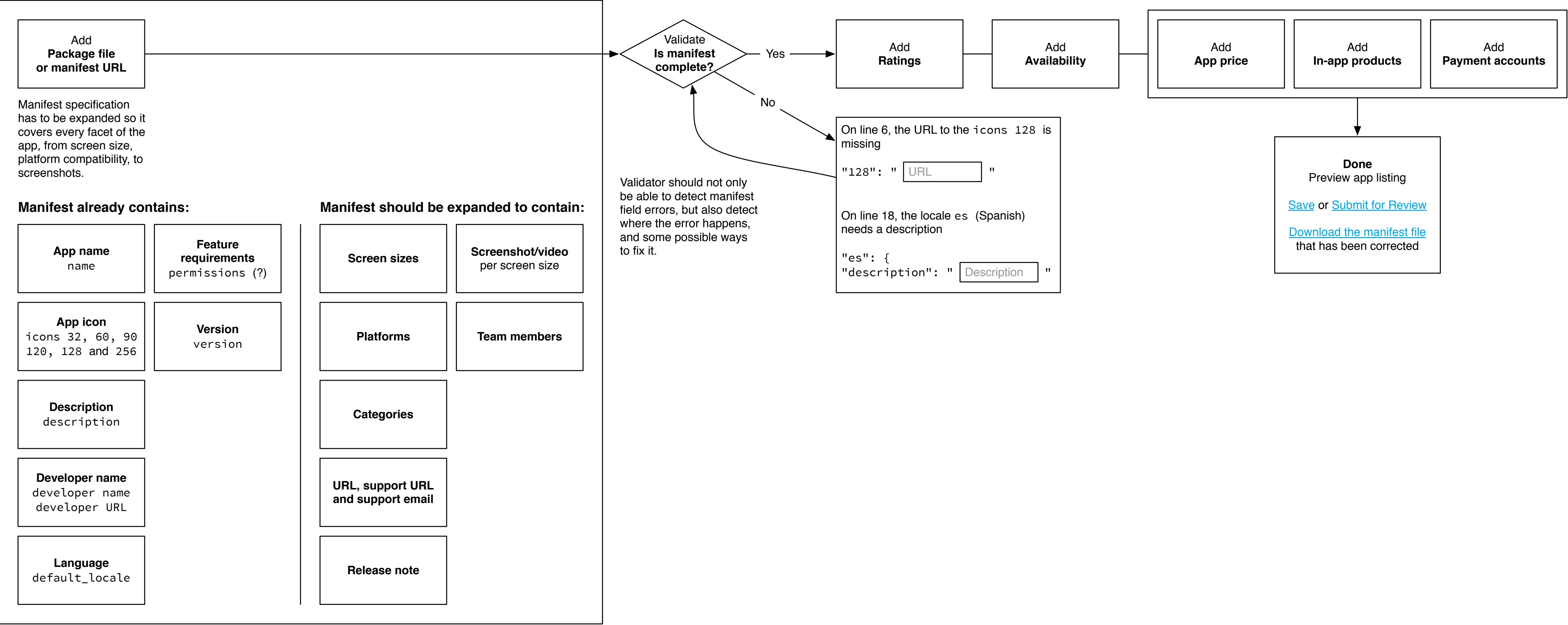
Flow idea #1

Base everything around a few basic questions. These questions will determine whether certain steps can be skipped later. Even though steps are skipped at creation, they will still appear on the editing IA as nodes. User can always change them later.
Pros: user can skip a lot of pages on submission. Cons: the creation steps does not mirror the editing IA.



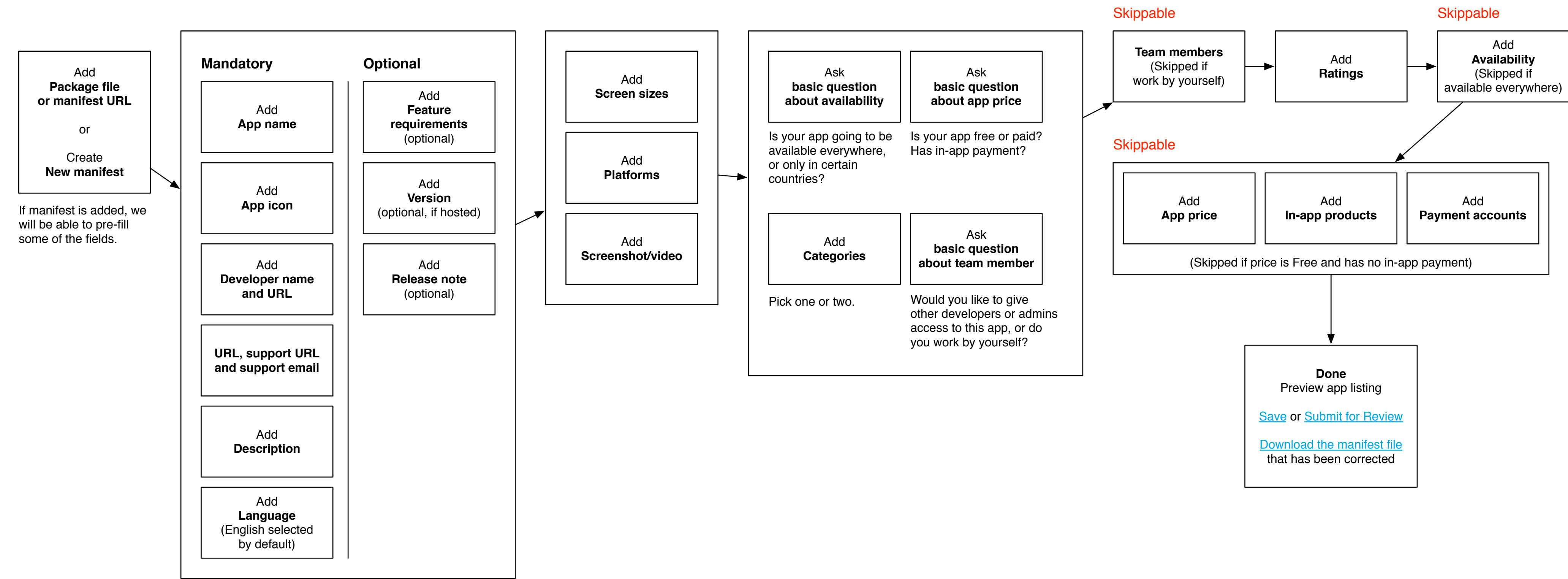
Flow idea #2: “manifest magic”

The idea here is for the manifest specification to be expanded to contain as much info about an app as possible, so that developers don’t have to fill out any forms. Even the correction to an error will happen in the manifest.



Flow idea #3: “manifest builder”

The idea is for people who has an app but feel that the manifest is daunting, to be able to go through the creation process without needing a manifest. As they go through the process, the system will build the manifest for them. In the end, they'll download a full manifest *and* submit an app, all at the same time.



Flow idea #4: “site sucker”

The idea is for developers who already have an app listed in another store to be able to instantly get listed in Firefox Marketplace.
Meaning, all they need to do is port the app binary (unnecessary if it’s a hosted web app), and we’ll take care of the listing.
Rather than adding a bunch of information, developers would only need to confirm whether the information we’ve got is correct.

