VSA  
Product Approach and Workflow

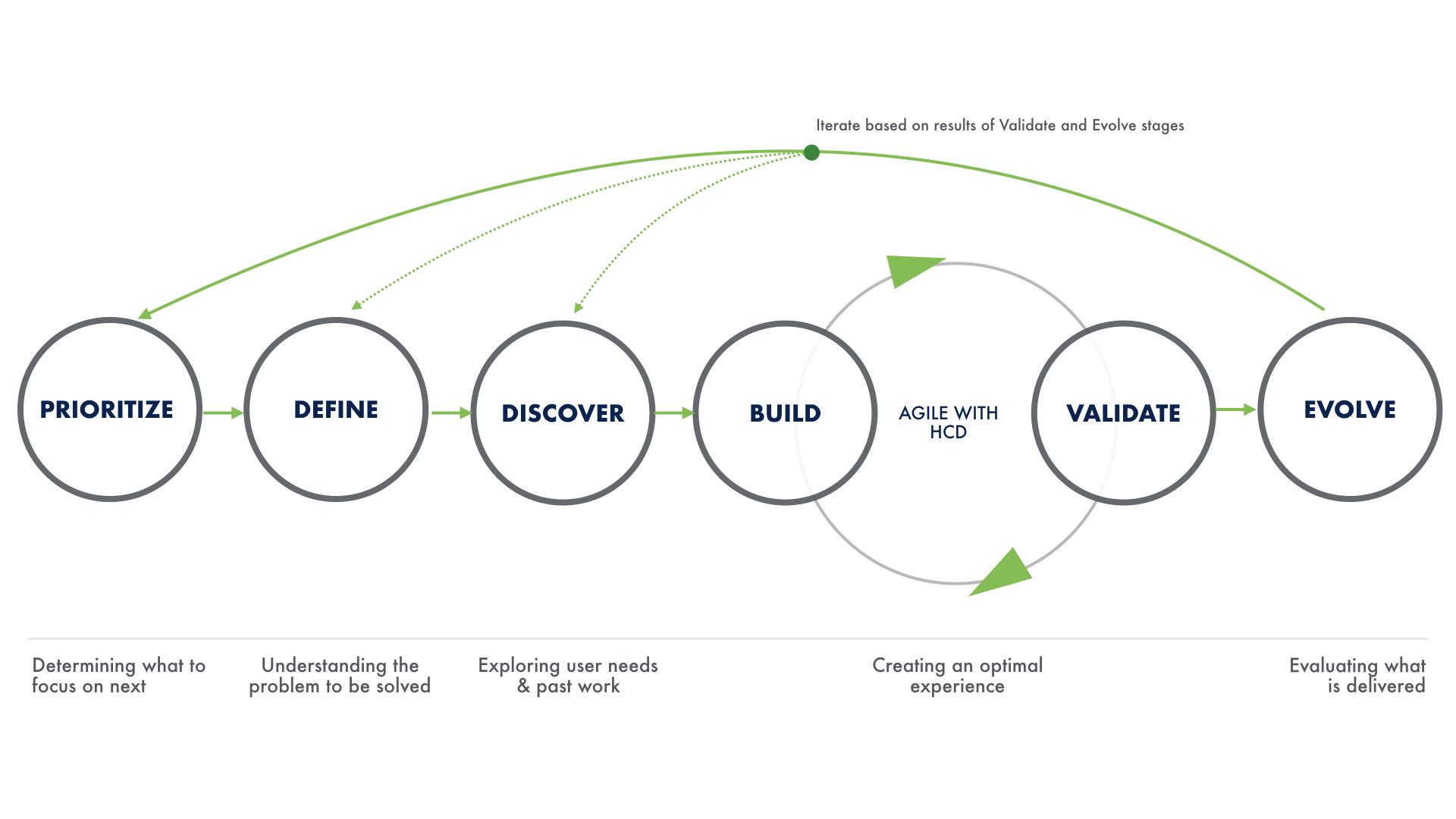
**DRAFT 2**

Authored by: Shawna Hein (VSA Design Lead) and Mickin Sahni (VSA Product Lead)

**Welcome!** This flow has been created to provide clarity for folks of all disciplines on how to follow a product process that integrates agile and human-centered design (HCD) methodologies.

In this process document, we follow a general cycle of a single request for a feature / work and go through 6 phases:

* [Prioritize](#_5fyel1v9yg48)
* [Define](#_l5iemq4re9m2)
* [Discover](#_bk21qme8maqz)
* [Build](#_hjptct35twxc)
* [Validate](#_1cmnzwhlsxxx)
* [Evolve](#_r0h5lhccbrdm)



Naturally, multiple requests have to be balanced with our program’s roadmap and timelines within the priorities of that roadmap. We would encourage thinking about each request in a kind of [gantt-chart](https://en.wikipedia.org/wiki/Gantt_chart) way, so the team can be in the “Discover” phase for one flow, but a “Build” phase for another.

This is also not a purely linear process, as seen in the diagram above. There may be times when the request has clearly met the expectations for the “definition of done” for a particular phase and can easily move on to the next phase. Additionally, most initiatives going through this flow will go through multiple rounds of Build/Validate before launch.

## Prioritize

This phase happens differently across different programs. Currently, on VSA, DSVA/DEPO leadership and PMs/POs all meet and decide on priorities for VA.gov. They then communicate these priorities to the PMs on each respective team. Currently, Chris Johnston (DSVA/DEPO) has told us that they take the following inputs when prioritizing:

* New policy
* Initiatives from leadership
* To some extent user feedback and help desk tickets

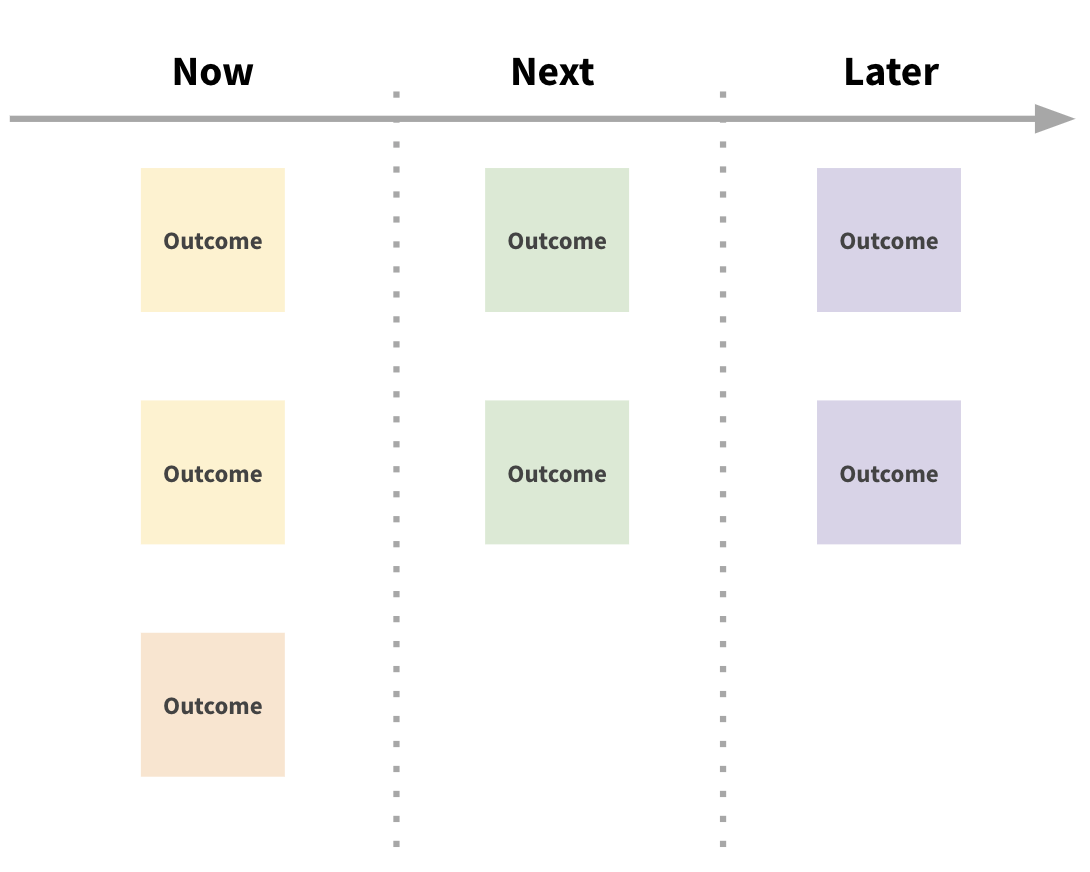
Ad Hoc / GCIO leadership (e.g. Luke, Mickin, Shawna, Jose) and the PMs on each team should be in communication with the DSVA/DEPO PMs/POs as much as necessary in this phase to avoid “handing down”, and ensure that user needs, data and analytics, and call center tickets are taken into account.

The goal of the Prioritize phase is to indicate the product outcomes we care about and how you’ll know when they’ve been achieved. These outcomes are prioritized against factors that help the team weigh competing objectives. These factors may include ROI and cost of delay or risk. The product team’s outcomes may be formatted as Objectives and Key Results (OKRs) and organized along, for example, a quarterly timescale or the more abstract and less prescriptive “Now, Next, Later”. These OKRs will be referenced later in the [Evolve](#_r0h5lhccbrdm) phase.

Once outcomes are prioritized, they move to the [Define](#_u5e8uobdp2yu) phase.

**Definition of done:** A roadmap of prioritized Objectives and Key Results that focus on the outcomes each product team seeks to achieve within a given period e.g. a quarter. As much as possible, the roadmap should avoid being feature or solution-oriented.

Template of a roadmap:



**Examples & Resources:**

1. [VA] [VSP Objective Roadmap](https://docs.google.com/presentation/d/1-8LlD8jljlv-C9IIQO1fHDC4ok0NzpJF3nLfwnx5610/edit#slide=id.g5bcedc037d_2_111)
2. [VA] [Sample Roadmaps](https://docs.google.com/presentation/d/1WrGz7ANEjDGhnCC1zsO3LI2vT1OZSkJdn5-xqxZQLbQ/edit#slide=id.g5fde8b3442_0_37)
3. [External] [Keep Features Off Your Roadmap](https://hackernoon.com/keep-features-off-your-roadmap-b14543340881?source=post_page-----28f8e1ff5406----------------------)
4. [External] [Folding Burritos 20 Product Prioritization Techniques: A Map and Guided Tour](https://foldingburritos.com/product-prioritization-techniques/)
5. [External] [Are You Doing OKRs Right?](https://www.range.co/blog/are-you-doing-okrs-right)
6. [External] [The Alternative to Roadmaps](https://svpg.com/the-alternative-to-roadmaps/)

## Define

The Define phase is where a team asks key questions to stakeholders to formulate a defined hypothesis towards solving user needs. A specific example of an outcome would be creating a product outline that includes a summary of where it stands, including historical notes on major decisions, how many users it has, and impact and performance metrics.

**Definition of done:** A loosely-defined hypothesis of the bet(s), under an initiative, which includes goals and outcomes for solving a specific user-types needs, that help inform the work created in the [Discover](#_bk21qme8maqz) phase. This usually results in some kind of Feature page or Project brief/outline.

**Flow**

1. An initiative has been defined as a priority via the team’s roadmap or communicated to the PM
2. PM/Team asks [key questions](#_iebtlkrz2xpr) below to create an initial project/feature brief and determine if initial [Discover](#_bk21qme8maqz) has been done, or is still needed.
   1. **If Discover has been done**, team leverages that research to create a clear definition of the problems and move towards [Build](#_hjptct35twxc) phase.
   2. **If not**, team moves forward with [Discover](#_bk21qme8maqz) to help inform that clear definition

**Examples & Resources:**

1. [VA] [Decision Review Requests and Appeals Digital Experience Outline](https://github.com/department-of-veterans-affairs/va.gov-team/tree/master/products/decision-reviews)
2. [VA] [Product Outline: eBenefits ("View My Rated Disabilities")](https://github.com/department-of-veterans-affairs/va.gov-team/blob/master/teams/vsa/teams/ebenefits/product-outline.md)
3. [VA] [Product Outline Template](https://github.com/department-of-veterans-affairs/va.gov-team/blob/master/platform/product-management/product-outline-template.md)
4. [VA] [21-526EZ Application for Disability Compensation and Related Compensation Benefits](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Disability/Disability%20526EZ/21-526EZ.md)
5. [External] [How to Write a Product Thesis to Communicate Customer Needs to Design and Engineering Teams](https://jasonevanish.com/2014/06/03/how-to-write-a-product-thesis-to-communicate-customer-needs-to-design-and-engineering-teams/)

## Discover

The Discover phase is where you learn the needs and pain points of our current and potential users, gain a deeper understanding of our stakeholders’ main concerns, get a handle on any existing solutions on legacy VA websites, and investigate any potential technical topics (e.g. run [technical “spikes”](https://scaledagileframework.com/spikes/)). The work you do during this phase will help inform everything that follows.

The methods used could be anything from user interviews with Veterans to a comparative analysis of our developer portal to engineering research around technical feasibility. An analysis of solutions on existing VA websites is also useful in shaping our approach. This is especially important as one of our objectives is to transition often disparate services to a single source of truth on VA.gov. All of these methods help us discover what the best path forward is for our team and our users.

There also may be times where another team has recent, relevant, and/or similar Discover on a particular topic that helps to inform our work. It’s important to recognize when we can leverage that information so that we are not duplicating efforts.

**Definition of done:** A researched definition of a bet has been identified with specific actionable items for our targeted user types. Epics are created to capture the work necessary for the whole product team and/or disciplinary teams to easily move forward to [Build](#_hjptct35twxc) a proposed solution.

**Flow:**

1. PM meets with team to discuss initiative and decide what [Discover activities](#_6tf0n5svp6hf) are needed (e.g. technical spike, qualitative interviews, etc)
2. Once Discover activities are established, PM writes tickets and assigns them to appropriate sprints. If working in a SAFe model, the PM may write Discover, Build, and Validate tickets for a feature for planning purposes.
3. Team members complete Discover tickets and communicate recommendations to the team. Any new problem areas uncovered or topics left unexplored during Discovery may be documented or captured in tickets for attention in the future.

**Examples & Resources:**

Overall

* [VA] DSVA’s [How to Run a Discovery Sprint](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/e558c4122b6c26c3f53fe6b9a6b0b95dda6aed4a/Practice%20Areas/Research/Discovery/DiscoverySprintHowTo.md)
* [VA] [Discovery sprint deliverables](https://github.com/department-of-veterans-affairs/vets.gov-team/tree/master/Products/Disability/Disability%20526EZ/discovery/2017-discovery-sprint)
* [External] [An Introduction to ZenHub Epics](https://help.zenhub.com/support/solutions/articles/43000010341-an-intro-to-zenhub-epics)

Research/Design Deliverables

* [VA] User research deliverable
  + [VSA] [Suggested Research Process](https://docs.google.com/document/d/1d2PJ6saIhzbWneevUs4rc153LTcbGxN-IJ9cXT51g1A/edit)
  + [VA example] Personalization team: Dashboard 2.0 [Conversation guide](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/4d386572b7170ae6d9058d2abd0ef11fb8e46c1e/Products/Identity/Personalization/Personalization%202.0/Discovery%20%26%20Research/Dashboard%20interviews/Conversation%20Guide.md), [Research plan](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/4d386572b7170ae6d9058d2abd0ef11fb8e46c1e/Products/Identity/Personalization/Personalization%202.0/Discovery%20%26%20Research/Dashboard%20interviews/Research%20Plan.md), [Research Summary](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Identity/Personalization/Personalization%202.0/Discovery%20%26%20Research/Dashboard%20interviews/Research%20Summary.md) (lead: Samara Strauss)
  + [VA example] Identity team: Accessibility study of login flow [Conversation guide](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Identity/Login/Research/508_Usability_Study/conversation_guide.md), [Research plan](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Identity/Login/Research/508_Usability_Study/research_plan.md), [Research summary](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Identity/Login/Research/508_Usability_Study/results_summary.md) (lead: Layla Soileau)
* [VA] Card sorting deliverable
  + [VA] Personalization team: [Card sort research plan](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Identity/Personalization/Personalization%202.0/Discovery%20%26%20Research/Card%20sort/Research%20plan.md), [Card sort summary](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Identity/Personalization/Personalization%202.0/Discovery%20%26%20Research/Card%20sort/Research%20Summary.md) (lead: Samara Strauss)

Product Deliverables

* [VA] Product Outline
  + <https://github.com/department-of-veterans-affairs/va.gov-team/blob/master/products/identity-personalization/personalization%202.0/Combine%20Profile%20and%20Account/README.md>
* [VA] Epics-Bets (Hypotheses)
  + [Address screen reader usability issues in the Sign In process](https://app.zenhub.com/workspaces/vft-59c95ae5fda7577a9b3184f8/issues/department-of-veterans-affairs/vets.gov-team/18101)
  + [Analytics Self-Service Implementation](https://github.com/department-of-veterans-affairs/va.gov-team/issues/2100)
  + [Decision Review Requests and Appeals Digital Experience: Board Appeal](https://github.com/department-of-veterans-affairs/va.gov-team/issues/2140)

## Build

The Build phase occurs once you’ve learned more about your stakeholders’ and users’ expectations so you can start creating testable outcomes. These activities and methods create artifacts that we can then [Validate](#_1cmnzwhlsxxx) with our users to see if they meet their expectations.

The Build phase can be building wireframes or other design artifacts, which we then Validate, and then move back to the Build phase to code and Validate again. Or if there isn’t time to Validate wireframes or prototypes, we may move directly to the coding stage and Validate once the feature is in staging. Build may also include development of API endpoints, integrations with other services, and application monitoring, which are necessary for the product to function.

Feature flags may be employed to help the team manage the test and release process of both front-end and back-end components.

**Definition of done:** A testable solution is created towards validating user expectations.

**Flow:**

If building design artifacts, e.g. wireframes:

* PM writes appropriate design tickets based on [Define](#_l5iemq4re9m2), [Discover](#_bk21qme8maqz), and/or [Validate](#_1cmnzwhlsxxx) work
  + This ticket should include any Discover findings necessary and any conversations that came about with others on the team as a result, e.g. any technical requirements the designer should keep in mind
* Designer picks up ticket and builds artifact
  + Note: at this stage, the designer should look at the [design topics checklist](#_vhc9875kfiyv) to make sure they’ve accounted for all necessary topics.
  + The designer is likely to come up with questions during this phase that need to be answered in order to create the best designs. At this point, the designer should be working with the PM to make sure these questions are answered and documented in a place visible to the whole team. We prefer some kind of matrix that includes the question, the answer, and an indication of who answered it and when.
* Once designer has some initial concepts, they should meet with VSP for a [design intent checkpoint](#_64vs4hopsvfx)
* Designer works with PM and team to determine if the artifact should then be Validated, or move directly to the coding part of the Build phase, depending on timeline and other factors.
* Once the designer has a final direction, they should get in a review with Shawna. Then they should [schedule a pre-development review with VSP](#_64vs4hopsvfx).

If building working code based on completed designs:

* PM writes appropriate tickets based on Discover work (or Validation work if this is following a Validation phase)
  + These tickets should always have the appropriate designer tagged.
  + It may help to also write necessary Validation tickets at this stage, unless you’re using a “Validate column” instead. For manual UI testing, tickets should be created at least 1 sprint before Validate Phase, with the **vsa-qa** label added -- see [Technical QA Checklist](#_x77lwexs9qjm) for details on what info should be included.
* PM refines (grooms) tickets with entire team (including DSVA PM) in Backlog Grooming
  + During backlog refinement (grooming), tickets are reviewed, updated, and broken-down
  + Tickets are re-prioritized to ensure they appropriately capture work necessary in the Build phase.
  + A/C should be reviewed for questions, and Validated for completeness / testability, blockers and dependencies should be identified
  + New tickets are created as needed to fill gaps in implementation
  + Designers and engineers should make sure they’ve put all needed info from their Discover efforts into the ticket, and, if needed, should look through the [Design Refinement checklist](#_1g054lvahf7s) to double check they’ve called everything out.
* PM develops [Sprint Objectives](https://github.com/department-of-veterans-affairs/va.gov-team/blob/master/teams/vsa/delivery/sprint-objectives/sprint-objectives-guide.md) and reviews these objectives with their DSVA counterpart and the team to ensure they align with VSA goals and initiatives
* PM leads the team through Sprint Planning with DSVA counterparts and the team
  + Team members review and commit to tickets that align with the sprint’s objectives
* Developer picks up ticket
* If it is something more than a small fix, the developer should reach out to the tagged designer (and potentially the PM) for a quick 30 minute “design meeting” to make sure they have a good understanding of how everything should work.
* Developer then codes the design or back-end component, reaching out to the designer, PM, or other developers on Slack when necessary.
* Once the ticket is coded, it moves to the [Validate](#_1cmnzwhlsxxx) stage

**Examples & Resources:**

Research/Design Deliverables

1. [[VA] Design Prototype](https://adhoc.invisionapp.com/share/DNQJQQ79JKB#/screens/347188092)
2. [VA] Design spec doc/page

Product Deliverables

1. [VA] [Combine Profile & Account Epic](https://github.com/department-of-veterans-affairs/va.gov-team/issues/2086)
2. [VA] [Migrate "Modify Dependents" from eBenefits to va.gov Epic](https://github.com/department-of-veterans-affairs/va.gov-team/issues/1976)
3. [VA] [Migrate "View Dependents" from eBenefits to va.gov Epic](https://github.com/department-of-veterans-affairs/va.gov-team/issues/1393)
4. [VA] [VSA Sprint Objectives Guide](https://github.com/department-of-veterans-affairs/va.gov-team/blob/master/teams/vsa/delivery/sprint-objectives/sprint-objectives-guide.md)

## Validate

The Validate phase is where we test (and re-test) our work automatically, manually, and with our users. This helps us build the best possible product. This phase includes methods such as usability testing and A/B testing, 508 Accessibility testing, as well as traditional technical QA.

NOTE: It is very important that time be given to the team to fix/iterate on things AFTER the Validate phase is done. This phase is pointless if there is no time given to iterate and improve afterwards. Therefore, Validate should almost always be followed up by another [Build](#_hjptct35twxc) phase of some sort, even if it is short.

**Definition of done:** A solution has been tested with users to help create the best product possible.

**Flow:**

If in the Validate phase with a design artifact, e.g. wireframes and prototypes:

1. Tickets or columns should be created for any usability testing or other validation activity that is done.
2. Once the designer/researcher runs the study, the results of the validation should be shared to the team
3. The team decides what issues to focus on and fix
4. Tickets are written to address what needs to be fixed, at which point the team is moved back to the [Build](#_hjptct35twxc) phase

in the Validate phase with a solution that has been built and pushed to staging:

1. Make sure you have some kind of solution for technical “validation” where each feature is put through a [technical QA checklist](#_x77lwexs9qjm) when needed and isn’t just left to whatever each person validating happens to remember / check
2. Make sure for each development ticket, there is a “sister” design QA ticket that is assigned to the appropriate designer (or some other method to make sure the appropriate designer Validates the ticket). If possible, there should be a way for the designer to view the work in an interactive environment so they can run through the [Design QA checklist](#_kijr23hs2uqv). A feature should not be pushed to production until the designer has QAed the feature.
3. After technical and design validation tickets are complete, there should be a short [Build](#_hjptct35twxc) phase where the developers fix any needed errors before pushing to production.

**Examples & Resources:**

Research/Design Deliverables

1. [VA] [Usability Testing Research Summary](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Identity/Personalization/Profile/Direct%20Deposit/Discovery%20%26%20Research/Research/Usability%20testing/Research%20Summary.md)
2. [VA] [Usability Study plan and results](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Health%20care/UrgentCare/Research/June-2019/readme.md)
3. Accessibility testing reports / tickets

QA Deliverables

1. [Manual QA testing matrix template](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Practice%20Areas/QA/QA_Testing_Matrix_Template.xlsx)

## Evolve

Evolve is a distinct feedback loop phase. This is where we learn how our product bets impact the Key Results established in [Prioritize](#_5fyel1v9yg48) and the KPIs we use to measure general health and success. These insights help the product team develop a strategy that incorporate lessons learned from previous iterations.

**Definition of done:** Bets are analyzed for their impact on product OKRs and KPIs. A readout, report, or presentation of the findings may be compiled with recommendations for how future iterations should proceed. Research readouts are a potentially helpful mental model of how an Evolve artifact might be structured.

**Flow:**

1. Product bets meet a threshold of completion e.g. time, adoption, usage.
2. OKRs and KPIs are evaluated before and after bets for changes and impacts
3. Findings and conclusions are documented via a report and/or retrospective
4. Recommendations are developed for how the team may proceed in subsequent iterations of [Prioritize](#_5fyel1v9yg48), [Define](#_l5iemq4re9m2), [Discover](#_bk21qme8maqz), and [Build](#_hjptct35twxc).

**Examples & Resources:**

* An artifact that captures key insights, recommendations, and next steps
* [VA] [Personalization 2.0 Discovery: Summary and Strategy](https://github.com/department-of-veterans-affairs/vets.gov-team/blob/master/Products/Identity/Personalization/Personalization%202.0/Discovery%20%26%20Research/Personalization%202.0%20Discovery%20Summary%20%26%20Strategy.md)

## 

## Further Details

### Define Phase

#### Key questions to answer

1. What problems are people facing?
2. What will people be able to do if you solve this problem?
3. What assumptions have we made about the problem and/or the user?
4. Why does the product or service exist?
5. How have we previously attempted to solve these problems?
6. Where does this problem fit in with other priorities?
7. Is there a particular time this needs to be completed? If so, why? What’s driving that timeline?
8. How do we know if we’re successful?
9. What issues might come up while working on this?
10. Who are your users?
11. What are they trying to do?
12. What tasks are they trying to accomplish? And how do they do that currently?
13. What problems or frustrations do they experience in trying to achieve their goals?
14. What do users need from your service to achieve their goals?
15. Do we currently have any analytics or help desk tickets about this issue?
16. If your research findings show a need for the service, how would you start building a technical solution?
17. Are there any aspects of a potential technical solution that needs further investigation?
18. What other systems or processes is this product or service dependent on?
19. Where is the data coming from?
20. Where is the data going to?
21. Are there other systems or processes that depend on this product?

### Discover Phase

#### Discover Activities

1. User interviews
2. A/B testing
3. Card sorting
4. Competitive analysis
5. Contextual Inquiry
6. Ethnographic research
7. Expert evaluation
8. Heuristic analysis
9. Mental modeling
10. Pluralistic walkthrough
11. Site audit
12. Sitemaps and content inventory
13. System mapping
14. User stories or scenarios
15. Technical Spikes

### Build Phase

#### Design Topics Checklist

1. Are the designs following accessibility best practices? Check with Jennifer/Trevor to make sure
2. Do the designs account for all edge cases, errors, and alternative states?
3. Do the designs account for responsive widths?
4. Do designs start mobile-first, to support Veterans that do not have computers?
5. Are the designs using design pattern library elements wherever possible?
6. If there are non-pattern library elements, is there a solid reason, is it defined clearly, and is it in-process for addition to the library?

#### Design Refinement Checklist

1. Are the appropriate designs attached / in a URL that the developers, QA, and PM can access?
2. Do we have URLs/destinations defined for all links?
3. Is there any interactive functionality that needs to be described?
4. Do we have sources for any new assets?
5. Which elements come from the design system?
6. Are there any new patterns that we should flag for inclusion into the design system?
7. Are there any dependencies on other flows getting into development first?
8. If this is a new page, what is the expected route?

#### Build artifacts

* Experience mapping
* Flow charts
* High-fidelity prototypes
* Mental modeling
* Mood boards
* Personas
* Scenario
* Sitemaps and content inventory
* System mapping
* User stories or scenarios
* User-Flow diagrams and navigation maps
* Wireframes

#### Interaction with VSP

The VSP team (Emily Waggoner is the Design Lead) is responsible for ensuring a consistent experience for the Veteran across various contracting teams and programs. Because of that, we need to include them in our design process as a resource, especially since we’re still relatively new to the VA and their design system. The expectation for interacting with VSP is the following:

##### For Medium - Large Projects

For these projects, we expect you to have 2 check-ins with VSP, one initially and one before development. VSP will provide some high level feedback during the meeting, then follow up with the rest of their feedback within 2 days. These two meetings will include VSP folks from design, content, IA, and accessibility.

1. **The Design intent checkpoint** - When you’re given a project or pick up a ticket, your PM should have already started some kind of feature brief doc with the basics of the project, the goals, questions and answers, etc. If you’re past discovery/user research, you can pick it up and come up with some initial thoughts/directions. I’d expect there to be a couple directions you’re considering. At that point, you should schedule a 30 minute design intent checkpoint with VSP (post the request in #vfs-platform-support and tag Emily Waggoner). You can also combine that meeting with presenting to your team, if you’d like to save time.
   * The design intent checkpoint will include:
   * Designer go over:
     1. What is the user story/problem we are designing for?
     2. Review whiteboard sketch of what you are imagining, or a rough screen flow
   * VSP will potentially provide:
     1. List of all relevant design components and patterns with appropriate URLs
     2. List of existing applications or pages with similar execution/UX flow, or patterns with URLs in staging, and relevant test user accounts
     3. Identification of any new design components, modifications to existing components, or
     4. Identification of any items to be added to a design component backlog
2. **The pre-development review -** When you’re confident of your design (and have reviewed it with Shawna, ideally), and think it’s ready for development, you should schedule a 30 minute review with VSP (post the request in #vfs-platform-support and tag Emily Waggoner). You can also combine that meeting with presenting to your team, if you’d like to save time.

The topics VSA will be looking at in the pre-development review include:

* New design patterns have been vetted by DSVA and documented
* Mobile, tablet, and desktop breakpoints have been accounted for
* Designs are accessible
* Correct use of design patterns in context
  + Form chapters in place
  + Privacy policy is present
  + Address form follows pattern if applicable
  + Name form follows pattern if applicable
* Correct use of typography
* Padding and spacing has been applied appropriately
* Form components used correctly in context
* Consistent use of iconography
* Colors use are consistent with color palette
* Primary and secondary CTA buttons used correctly in context
* :hover and :focus states have been accounted for and are consistent with design patterns
* Alert messages have been used appropriately
* Error states have been accounted for and used appropriately
* Loading indicators have been used where appropriate

##### For Small Projects

If you’re working on a small ticket that is only a few days of work, you don’t have to go through the whole review process above. Instead, if you feel like you need a little help or guidance on design elements or copy, please attend VSP’s office hours on Tuesdays at 3pm EST ([copy the event to your calendar here](https://calendar.google.com/event?action=TEMPLATE&tmeid=Nmx2aW5rczVhZXB0bTJwcTNoc2Z2ZGRnN2JfMjAxOTEwMjJUMTkwMDAwWiBlbWlseUBhZGhvY3RlYW0udXM&tmsrc=emily%40adhocteam.us&scp=ALL)), or message them in their slack channel, #vfs-platform-support

### Validate Phase

NOTES:

* Please give adequate notice (min. 1 sprint) to QA for scheduling manual testing.
* Staging test-data must be set up behind API-endpoints for each test-user and user-flow/-scenario. Unlike local environments where most API responses are locally mocked, Staging retrieves test-data from the actual API-endpoints.
* Due to general QA-resource restrictions and potential conflicts with other launches, designers/developers may need to be recruited to participate in manual browser/device testing.

#### Technical QA Checklist

1. Are Design comps for ALL user stories/scenarios (incl. errors/alerts) provided to QA?
2. Are Staging test-users identified for ALL user stories/scenarios of the feature?
3. Does the feature pass manual UI (visual & interactive) testing for all supported browsers/devices?
   1. Feature visually matches design comps?
   2. Copy matches design comps (or copy specs if separately provide)?
   3. Any broken links (incl. references to info/resources outside the feature)?
   4. Interactive functionalities produce expected results?
4. Does the feature pass 508 Accessibility testing?  
   [Note: Contact Ad Hoc Accessibility Designers Jennifer Strickland (VSA) and Trevor Pierce (VSP) for actual testing -- QA team currently do not have licenses for specialized screenreaders.]

#### Design QA Checklist

1. Check both Authenticated and Unauthenticated if relevant
2. Interactions - make sure interactions work as described
3. Design system - make sure all elements that need to use design system elements do so
4. Spacing, Padding & Alignment
   1. Make sure things are consistent
   2. Check for vertical alignment
5. Responsiveness
   1. Check small, medium, large, and extra-large widths and make sure everything looks the way it’s supposed to
   2. Check on as many devices as you have the ability to as emulators may render inaccurately
6. Links
   1. Make sure links go to the correct places
   2. Check that any link labeling conventions are applied correctly
7. Check edge cases / errors and make sure they display as designed
8. Headings - make sure they’re the correct style / hierarchy / size
9. Copy - verify that the correct copy has been used and there are no typos or other errors
   1. We won’t always be responsible for this hopefully.