

# Golden Flow Designs

## Golden Flow Design: Accelerating Discovery Through Rapid Iteration

"Focus on what matters, test often, and build better products faster."

### Possible outline and organization of the article

"We've all been on *that* project. The one where requirements are vague, stakeholders can't agree on the product's vision, and it feels like every discussion goes in circles. You're left wondering how to move forward when nothing seems clear."

"This is exactly where the Golden Flow technique comes in. It's a simple, pragmatic approach to cut through the noise, focus on what matters most—the user's primary goals—and accelerate the discovery process."

I work in IT automation, a product space that requires both technical and business domain-level understanding. As a designer in this field, I often cannot be an expert in the product domain. Therefore, I rely heavily on my process and methodology. Working in such spaces necessitates bridging understanding and quickly ramping up on just enough knowledge as it becomes necessary, especially when dealing with technical, enterprise-grade products.

### The Problem

Many designers create product features with immediate detail, even though they lack a thorough understanding of the product or its underlying technology. These early-stage efforts often involve untested assumptions that need validation. With the advent of large design systems, designers now have access to a

toolkit of highly detailed components. While useful in later stages, I argue that using these polished components too early in the process hinders rapid, iterative product discovery.

"In this article, we'll explore how Golden Flows can serve as your north star in early-stage product design. By the end, you'll walk away with a practical approach you can incorporate into your own work, helping you align teams and create a clear user-focused direction even when requirements are uncertain."

## What is a Golden Flow?

"A Golden Flow represents the primary journey a user takes through an application to achieve their main goal. It's the most used, most recognized path within the product—the 'golden thread' that ties the user's actions to their expected outcome. While a product may offer many user journeys, the Golden Flow focuses on the core experience, leaving out secondary use cases or edge scenarios to maintain clarity and focus."

Why - what does it solve

## Real-World Example: Golden Flow on a Travel Site

Let's imagine a travel site called **SkyRoutes**, which specializes in selling flights. The primary user goal is to purchase a flight, and these might be the steps they go through:

- **Search for flights** (location/s, time/date, passengers)
- **View available flight options** (airline, time, layover)
- **Select flight** (fare, class)
- **Flight options** (bags, seats)
- **Review**
- **Purchase**

The primary user goal is to travel, and the task is to find a flight. This Golden Flow represents the core journey for a typical user, guiding them through each essential step to achieve their

objective. It assumes the user seeks a straightforward one-way or round-trip flight, focusing on clarity and efficiency.

While there are more complex scenarios, such as multi-city flights or additional travel packages, along with other workflows like trip planning or account creation, those would fall outside this primary flow. The Golden Flow keeps the focus on the core user task to deliver the most seamless experience possible.

---

### 3. Why Golden Flows

#### 3.1 Focusing on the Core Journey

- Starting with the core journey helps center attention on the primary purpose of the product.
- It ensures that the most critical concept is addressed first, forming a foundation for extending to other use cases and flows.

#### 3.2 Building Understanding

- By prioritizing the Golden Flow, teams gain a better understanding of the user, their problem, and the business objectives before tackling secondary features or extras.

#### 3.3 Preventing Distraction

- Golden Flows keep the team aligned, avoiding extraneous details and preventing random exploration of less relevant journeys.
- It follows the 80/20 rule: focus on the primary path first to deliver maximum value efficiently.

#### 3.4 Confirming Opportunity and Hypothesis

- Use the Golden Flow to confirm market opportunity and validate your primary hypothesis.
- This ensures downstream work is built on a solid, user-centered foundation.

#### 3.5 Iterative Refinement

- Begin with rough wireframes that block out the core steps.
- Collaborate with team members, subject matter experts, and users to iteratively refine and add details.

#### 3.6 Testing Assumptions

- As the Golden Flow evolves, use it as a storyboard or prototype to test assumptions.

- This provides a rapid, efficient way to clarify product intent and outcomes.
- 

## 4. Collaborating to Define Golden Flows

To create effective Golden Flows, you'll need input from two key perspectives:

1. **End Users** – Understand their needs, challenges, and pain points. Look for insights like "I wish we could do X" or "We spend too much time on Y."
2. **The Business** – Partner with product managers, marketing teams, and subject matter experts to align with business goals and user feedback.

By combining these insights, you'll develop a user-centric flow that balances user needs with business objectives. Once defined, validate the result with typical users to ensure it resonates and delivers value.

---

## 5. Validating the Golden Flow

### 5.1 Creating Minimal Artifacts

- Develop low-resolution artifacts, such as rough wireframes, that capture the Golden Flow.
- Socialize these artifacts with stakeholders to spark internal debate and gather feedback.

### 5.2 Micro-Iterations

- Use customer feedback to refine and iterate on these artifacts.
  - Progressively increase resolution, evolving from rough concepts to testable prototypes that can validate the Golden Flow in real-world scenarios.
- 

## 6. Breaking Down the Golden Flow Technique

### 6.1 Core Principles

- Focus on the user's main goal.
- Emphasize simplicity and adaptability.

## 6.2 Step-by-Step Process

- How to identify and define Golden Flows.

## 6.3 Tools and Frameworks

- Tools and methods that complement this technique.
- 

## 7. Case Study or Practical Example

### 7.1 Realistic Use Case

- Demonstrating the application of Golden Flow design.

### 7.2 Impact

- Simplicity, clarity, and alignment across teams.
- 

## 8. Takeaways

### 8.1 Practical Advice

- Tips for UX designers on starting with Golden Flows.

### 8.2 Reinforcement

- Highlight the role of Golden Flows in bootstrapping projects and keeping the user focus.
- 

## 9. Conclusion

### 9.1 Recap

- Summarize the value of Golden Flows.

### 9.2 Encouragement

- Invite readers to apply the concept and share their experiences.
- 
- 

## Why I'm Writing This Article

## The Issues with High Fidelity Too Soon

When designers jump straight into creating detailed mockups using polished design system components, they encounter several issues:

### 1. Mismatched Process Stage and Fidelity

- High-fidelity designs prematurely signal a confidence in design decisions that may not yet be validated.

### 2. Distractions During Product Reviews

- High-detail designs often lead stakeholders to focus on minor issues like labels or typos, distracting from the overall product concept.

### 3. Inaccurate Representation of Understanding

Early-stage designs should reflect the exploratory nature of the process, but high fidelity misrepresents this, creating confusion about the maturity of the product.

## Matching Fidelity to Understanding

Early-stage ideas should be rendered with minimal detail, using only what's necessary to communicate the concept. This ensures:

- Less distraction during reviews.
- A clear visual indication that the design is in its exploratory phase.

For example, early designs could use grayscale or intentionally rough visuals, akin to how car designers use clay models to differentiate between concept and final product. By doing so, designers make it clear that these are preliminary explorations – "clay," not the finished product.

## Two Toolkits for Designers

Designers should operate with two distinct toolkits:

### 1. Low-Fidelity Toolkit

- For early Golden Flow design.
- Focuses on minimal detail and emphasizes the product experience.

## 2. Standard UI Component Toolkit

For later stages when the core experience has been validated and refined.

### The Importance of Iteration

At the start, the goal isn't about perfecting UI design—it's about nailing the product experience. Design is a process of theory-building based on testing assumptions. Each iteration brings new learnings, some confirming previous ideas and others offering fresh insights.

UI design should follow once the foundational Golden Flow for the product experience is correct. Spending too much time on insignificant details in the early stages reduces the time available for feedback and iteration. I directly associate a product's success with the number and quality of iterations informed by real-world feedback.

### Golden Flow Design as Iterative Knowledge Building

Golden Flow Design is about building iterative knowledge, much like the concept of "Programming as Theory Building." You start with the bones of the idea, gather feedback, iterate, and progressively add detail and resolution as the product matures. This method ensures the product evolves deliberately, balancing exploration and refinement to create a cohesive and effective user experience.

---

---

---

I'm still getting organized on my thoughts. I want designers to leave with this as a method of accelerating the design process by a rapid and iterative approach to product discovery. In many projects, designers tend to think "strategic design" up front, which can conflict with agile methodology of start small, build and iterate incrementally while learning vs big design up front.

At the heart of this method is ensuring the designer understand the users' golden flow, or primary path to achieve the top level goal. This helps focus the design. Rather than tackling lesser important secondary and tertiary flows and use cases, this method sets the backbone for iterative rounds of "micro-refinements". Starting first with the user, the context of the situation, the original goal, and tasks needed to complete the goal, designers iterate numerous times in this process, every round adding detail and resolution, each round being sufficiently tested or gaining input from users, so that the designer does not make false assumptions early in the project, by going directly into detailed design too early, or tackling user flows that are not critically essential.

I'm trying to come up with headlines for my UX software article focused on accelerating the discovery process by rapid iteration through golden flows and low-fidelity design

I want to highlight to use this process to test high level product assumptions, the use case, user value, the concept of the feature, not the actual UI. This method suggests low fidelity wireframing, testing with iterative increase of details, resolution and fidelity. And by doing this process, one can accelerate outcomes and the outcomes would be of higher quality due to the frequency and amount of feedback / improvement loop.  
I'm going to use this canvas to try and collect all the various thoughts

### Core problem

This is a core problem, the reason for adopting a process like this.

The idea of this is to use the design process to make the abstract idea as concrete, as soon as possible. The idea in this process isn't about make the UI, or even the ux. Its broader, its about taking vague ideas, and often times mission requirements, missing pr/faq. Its often left at the behest of the designer to start making artifacts. Without visuals, people often misunderstand each other, think they are aligned, but without a concrete picture, often have different interpretation of the documents, words, conversation and all the variuos method of trying to define software

---

### ***Enter Golden flow design....***

What Golden flows are

Concrete example

Process of flow stitching to connect user, steam sna dflows

### **the Wire framing method**

This section describes the process of starting simple, staying in low fidelity with minimal details, then the stages of increasing in fidelity

Ideas on how wireframes help in early stage concept

development

- Start rough, low detail, then progressively add as more is known
- It helps keep the focus on flow, going broad and shallow across task completion in the journey
- It communicated the level of understanding. The fidelity matches the understanding.
- Enables other to add in, to contribute, to help fill in the details
- Keeps stakeholders focused on what's important
- Iterations allow details like content and heirchy
- Hierarchy of content and features.
- Information architecture, verbiage, and copy definition.

- Process and interaction flow/sequences.
- Rapidly test new ideas and validate them with users.

ChatGPT said:

ChatGPT

## The Value of Low-Fidelity Wireframes in the Design Process

### Wireframes as a Starting Point

Wireframes serve as a foundational tool in the early stages of concept development. By starting rough and keeping detail levels low, they encourage rapid ideation and flexibility. As understanding of the product or feature grows, the wireframes evolve—progressively incorporating more detail to reflect the team's deepening insights.

### Focus on Flow Over Details

Low-fidelity wireframes prioritize the user journey and task flow over intricate design elements. This broad-and-shallow approach ensures that critical pathways are explored early, facilitating holistic thinking about how users will interact with the product to achieve their goals.

### Communicating Understanding

The level of fidelity in wireframes often mirrors the team's level of understanding. Early-stage sketches may be sparse and abstract, signaling areas where discovery is still needed. This transparency allows the team and stakeholders to align on what is known versus what needs exploration.

### Fostering Collaboration

Wireframes act as a shared canvas, inviting contributions from diverse perspectives. Whether it's refining processes, clarifying interaction sequences, or adjusting content hierarchy, wireframes create a space for stakeholders to co-create and fill in gaps together.

### Keeping Stakeholders Aligned

In early stages, there's a tendency for stakeholders to focus on aesthetic details or tangential concerns. Low-fidelity wireframes mitigate this risk by keeping attention on the essentials: user

flows, task completion, and overall structure. By avoiding premature focus on visual design, the team remains grounded in solving the right problems.

### Iterative Detailing

Wireframes enable a layered approach to design. Initial iterations focus on major components like information architecture, verbiage, and interaction sequences. Over time, details like content hierarchy and copy can be added, ensuring the design grows organically alongside deeper product understanding.

### Testing and Validation

One of the greatest strengths of wireframes is their adaptability for testing. Their simplicity allows teams to prototype and test ideas rapidly, gathering valuable feedback to validate assumptions. This reduces the risk of overinvesting in ideas that might not resonate with users or meet their needs.

By embracing low-fidelity wireframes, teams can maintain agility, encourage collaboration, and focus on delivering a user-centered product that evolves with both discovery and clarity.

---

### *This is how the process works...*

This is how it works. Starting with a basic definition of the user, let's use an example of a Site Reliability Engineer. Their primary goal is to maintain the reliability, stability, security and availability of assigned workloads. In doing such jobs, a golden flow might include:

The user - Site Reliability Engineer

Key tasks they do:

**Incident resolution** - detecting issues, tracing the issue, finding the root cause analysis, fixing and documenting for future scenarios

**Proactive remediation** - identifying and enabling resources to detect and predict future problems/failures, then provide automations that mitigates these issues.

In this method, I generally advise the designer to start with a well written user scenario description which includes, the user type, what they do, context of the situation, the goal of the tasks and steps they do to accomplish this, their expected outcome, and measures that quantify the success or failure of the "golden flow"

***So, the method is:***

Write down the basics in a Word file or something similar

Create a basic flow diagram with the steps and other annotations needed

Review with SMEs and stakeholders. Get feedback, iterate, improve

Develop a low-fidelity storyboard that captures the essential steps. Stay lo-fi, minimal, only essential details (headline, key sub headings or minimal text) Greek or block the rest. The point here is capture minimal content to keep a. designers from wasting time on content, b. to keep stakeholders from tuning on unnecessary detail.

Storyboard can be in pencil or rough wires. The key is to be representative of the steps, not the actual UI. You are trying to target getting Story sequence correct first, then the number of steps in the journey, then start adding content, and lastly adding the UI layer to it

What makes this process unique is 1) the number of feedback loops you can get in ( more than if you just 'design' it and user test it. 2) it builds understanding and confidence each loop of iteration in the process. It's kind of like that Computer science article "Programming as theory building"

With this method, you start with a base set of understanding and theory, and at each round of feedback and iteration, you learn more, get more confident in your learnings.

## The Problem

Many designers create product features with immediate detail, even though they lack a thorough understanding of the product or its underlying technology. These early-stage efforts often involve untested assumptions that need validation. With the advent of large design systems, designers now have access to a toolkit of highly detailed components. While useful in later stages, I argue that using these polished components too early in the process hinders rapid, iterative product discovery.

### **The Issues with High Fidelity Too Soon**

When designers jump straight into creating detailed mockups using polished design system components, they encounter several issues:

#### Mismatched Process Stage and Fidelity

High-fidelity designs prematurely signal a confidence in design decisions that may not yet be validated.

#### Distractions During Product Reviews

High-detail designs often lead stakeholders to focus on minor issues like labels or typos, distracting from the overall product concept.

#### Inaccurate Representation of Understanding

Early-stage designs should reflect the exploratory nature of the process, but high fidelity misrepresents this, creating confusion about the maturity of the product.

#### Matching Fidelity to Understanding

Early-stage ideas should be rendered with minimal detail, using only what's necessary to communicate the concept. This ensures:

Less distraction during reviews.

A clear visual indication that the design is in its exploratory phase.

For example, early designs could use grayscale or intentionally rough visuals, akin to how car designers use clay models to differentiate between concept and final product. By doing so, designers make it clear that these are preliminary explorations—"clay," not the finished product.

## Two Toolkits for Designers

Designers should operate with two distinct toolkits:

### Low-Fidelity Toolkit

For early Golden Flow design.

Focuses on minimal detail and emphasizes the product experience.

### Standard UI Component Toolkit

For later stages when the core experience has been validated and refined.

## The Importance of Iteration

At the start, the goal isn't about perfecting UI design—it's about nailing the product experience. Design is a process of theory-building based on testing assumptions. Each iteration brings new learnings, some confirming previous ideas and others offering fresh insights.

UI design should follow once the foundational Golden Flow for the product experience is correct. Spending too much time on insignificant details in the early stages reduces the time available for feedback and iteration. I directly associate a product's success with the number and quality of iterations informed by real-world feedback.

## Golden Flow Design as Iterative Knowledge Building

Golden Flow Design is about building iterative knowledge, much like the concept of "Programming as Theory Building." You

start with the bones of the idea, gather feedback, iterate, and progressively add detail and resolution as the product matures. This method ensures the product evolves deliberately, balancing exploration and refinement to create a cohesive and effective user experience.

---

## The Value of Low-Fidelity Wireframes in the Design Process

### Wireframes as a Starting Point

Wireframes serve as a foundational tool in the early stages of concept development. By starting rough and keeping detail levels low, they encourage rapid ideation and flexibility. As understanding of the product or feature grows, the wireframes evolve—progressively incorporating more detail to reflect the team's deepening insights.

### Focus on Flow Over Details

Low-fidelity wireframes prioritize the user journey and task flow over intricate design elements. This broad-and-shallow approach ensures that critical pathways are explored early, facilitating holistic thinking about how users will interact with the product to achieve their goals.

### Communicating Understanding

The level of fidelity in wireframes often mirrors the team's level of understanding. Early-stage sketches may be sparse and abstract, signaling areas where discovery is still needed. This transparency allows the team and stakeholders to align on what is known versus what needs exploration.

### Fostering Collaboration

Wireframes act as a shared canvas, inviting contributions from diverse perspectives. Whether it's refining processes, clarifying interaction sequences, or adjusting content hierarchy, wireframes create a space for stakeholders to co-create and fill in gaps together.

### Keeping Stakeholders Aligned

In early stages, there's a tendency for stakeholders to focus on

aesthetic details or tangential concerns. Low-fidelity wireframes mitigate this risk by keeping attention on the essentials: user flows, task completion, and overall structure. By avoiding premature focus on visual design, the team remains grounded in solving the right problems.

### Iterative Detailing

Wireframes enable a layered approach to design. Initial iterations focus on major components like information architecture, verbiage, and interaction sequences. Over time, details like content hierarchy and copy can be added, ensuring the design grows organically alongside deeper product understanding.

### Testing and Validation

One of the greatest strengths of wireframes is their adaptability for testing. Their simplicity allows teams to prototype and test ideas rapidly, gathering valuable feedback to validate assumptions. This reduces the risk of overinvesting in ideas that might not resonate with users or meet their needs.

By embracing low-fidelity wireframes, teams can maintain agility, encourage collaboration, and focus on delivering a user-centered product that evolves with both discovery and clarity.

---

### Random notes

This is a core problem, the reason for adopting a process like this. The idea of this is to use the design process to make the abstract idea as concrete, as soon as possible. The idea in this process isn't about make the UI, or even the ux. Its broader, its about taking vague ideas, and often times mission requirements, missing pr/faq. Its often left at the behest of the designer to start making artifacts. Without visuals, people often misunderstand each other, think they are aligned, but without a concrete picture, often have different interpretation of the documents, words, conversation and all the variuos method of trying to define software

I work in IT Automation, a product space that requires both technical and business domain level understanding. As a designer, when I work on these products, I often cannot be an expert in the product domain, so I have to rely on my process and methodology. When working in these spaces, it's essential to bridge understanding, to quickly ramp up on just enough knowledge at the time it's needed, especially in technical, enterprise products. The issue I see is designers creating product features immediately with detail, even though they don't have a great understanding of the product or technology, and they've made early assumptions that need to be tested.

With the advent of large design systems, designers have been given a tool kit of lego parts with details and fidelity that they just don't need in early stage product discovery. Actually to the contrary, I would argue that level of finality and polish in utilizing design system components is a detriment, an obstacle to rapid and iterative product discovery. Here's why, designer takes a spec at face value and starts building a mockup. They pull all the components from their company's design system. The mockup looks great, the designers love the polish and how hot it looks. Here's the issue. First, it inaccurately represents where you are in the stage of the process. Being in high fidelity communicates a confidence of design decisions. Detail provides points of distraction in product reviews. Ever notice how stakeholders can nitpick a label or a misspelling, and miss the overall product idea. So in this case, the fidelity does not accurately reflect the understanding. I would argue that we match fidelity to understanding and stage. Early ideas should be rendered with the least amount of detail needed. No more detail than absolutely needed. Less to get distracted on.

Everything is in grayscale. Make it very intentional that this is clay, not the final product. When car designers model cars, they use clay - no chance to confuse model of a product, and final product. So, designers should have two toolkits, a low fidelity toolkit for early Golden flow design, then their standard UI component design system. I can't stress this enough. At the start, it's not about the UI design, it's about getting the product experience correct. Design is a process of theory building based

on tested assumptions. With each iteration come new learnings, some confirmation, some new learning. UI design can come later in the work, when the basic golden flow for the product experience is correct. I've seen so many designers waste time on insignificant detail, spend way too long, reducing the time to get feedback, and iterations. I'd directly associate the success ratio of a product with the iterations from the feedback of the product design. Golden flow design is like building iterative knowledge, similar to "Programming as theory building", where you get the bones out there, and you get feedback and you iterate and increase detail and resolution along the journey of final maturity.

---

---

New section on value of lo-fidelity and wireframing in the process. Let's keep these thoughts in mind: Ideas on how wireframes help in early stage concept development - Start rough, low detail, then progressively add as more is known - It helps keep the focus on flow, going broad and shallow across task completion in the journey - It communicates the level of understanding. The fidelity matches the understanding. - Enables others to add in, to contribute, to help fill in the details - Keeps stakeholders focused on what's important - Iterations allow details like content and hierarchy - Hierarchy of content and features. - Information architecture, verbiage, and copy definition. - Process and interaction flow/sequences. - Rapidly test new ideas and validate them with users.

---

---

Purpose of article My intent was for projects you get to work on, but you really aren't familiar with either the product space (

think some abstruse software to do inventory management ), or the technical space is outside of your area of expertise. In either or both cases, you could be faced with what what i refer to as airliner cockpit, you see dials and knobs, have no idea of what they mean, or how the user interacts with them to accomplish their goals. This situation, you are essentially making parts, but don't understand the product, thus you might be making it worse.

Actually, I might have a different angle, it might be the real outcome I was seeking. Agile and design systems thinking are at odds. One approach starts small, avoids excessive up front design through fast cycles of micro improvements and adjustments, where as design systems thinks globally, forcing big up front design. It like build what you know and foresee now, vs deliberate plan and debate. So my technique, the method I use is called golden flow design.

Golden flow design is purposeful and relentless focus on the core path, the reason for the product, its 80 percent use case. The problem is that teams tend to spend on feature or product features all too easily - not the overall job and journey. That's not to say that it is not needed (feature level design - like time or calendar picker on a scheduling flow), but often designers are just working on the micro-specific, and just seeing their own specific silo of work, not considering or being able to see work and steps that connect together. And because of that lack of adjacent visibility, context and understanding, designers can craft parts of a product experience that work well isolated, but don't blend together as intended. As a result of that core experiences for the end user completing a job to be done, the product can appear to be disjointed when the collective set of puzzle piece flows are added into a singular whole. I wonder if i should delineate golden flow ( the primary flow ), vs secondary and tertiary flows that are needed for a product. All are needed.

Now, don't misunderstand me and interpreting that there is only one primary path. I think its more about altitude and

resolution. For example, a travel site. At the 30,000 foot level, the job to be done at that level is the fulfillment of travel to a destination for a traveler - details are out of focus so the resolution excludes information like type of transportation, membership, payment type - those kind of details. So at a very high level, course resolution, the golden flow of the site is search and find

---

---

Why i'm writing this article I work in IT Automation, a product space that requires both technical and business domain level understanding. As a designer, when i work on these products, I often cannot be an expert in the product domain, so i have to rely on my process and methodology. When working in these spaces, its essential to bridge understanding, to quickly ramp up on just enough knowledge at the time its needed, especially in techincal, enterprise products. The issue I see is designers creating product features immediately with detail, even though they don't have a great understanding of the product or technology, and they've made early assumptions that need to be tested. With the advent large design systems, designers have been given a tool kit of lego parts with details and fideitly that they just don't need in early stage product discovery. Actually to the contrary, I would argue that level of finality and polish in utilizing design system components is a detriment, an obstacle to rapid and iterative product disocvery. Here's why, designer takes a spec at face value and starts building a mockup. They pull all the components from their company's design system. The mockup looks great, the designers love the polish and how hot it looks. Here's the issue. First, it inaccurately represents where you are in the stage of the process Being in high fidleity communnicates a confidence of design decisions Detail provide points of distraction in product reviews. Ever notice how stakeholders can nit pick a label or a

mispelling, and miss the overall product idea. So in this case, the fidelity does not accurately reflect the understanding. I would argue that we match fidelity to understanding and stage.

---

---

I want designers to leave with this as a method of accelerating the design process by a rapid and iterative approach to product discovery. In many projects, designers tend to think "strategic design" up front, which can conflict with agile methodology of start small, build and iterate incrementally while learning vs big design up front. At the heart of this method is ensuring the designer understand the users' golden flow, or primary path to achieve the top level goal. This helps focus the design. Rather than tackling lesser important secondary and tertiary flows and use cases, this method sets the backbone for iterative rounds of "micro-refinements". Starting first with the user, the context of the situation, the original goal, and tasks needed to complete the goal, designers iterate numerous times in this process, every round adding detail and resolution, each round being sufficiently tested or gaining input from users, so that the designer does not make false assumptions early in the project, by going directly into detailed design too early, or tackling user flows that are not critically essential. Starting with a basic definition of the user, let's use an example of a Site Reliability Engineer. Their primary goal is to maintain the reliability, stability, security and availability of assigned workloads. In doing such jobs, a golden flow might include: Incident resolution - detecting issues, tracing the issue, finding the root cause analysis, fixing and documenting for future scenarios Proactive remediation - identifying and enabling resources to detect and predict future problems/failures, then provide automations that mitigate these issues. In this method, I generally advise the designer to start with a well-written user scenario description which includes, the user

type, what they do, context of the situation, the goal of the, tasks and steps they do to accomplish this, their expected outcome, and measures that quantify the success or failure of the "golden flow" So, the method is: Write down the basics in a Word file or something similar Create a basic flow diagram with the steps and other annotations needed Review with SMEs and stakeholders. Get feedback, iterate, improve Develop a low-fidelity storyboard that captures the essential steps. Stay lo-fi, minimal, only essential details (headline, key sub headings or minimal text ) Greek or block the rest. The point here is capture minimal content to keep a. designers from wasting time on content, b. to keep stakeholders from tuning on unnecessary detail. Storyboard can be in pencil or rough wires. The key is to be representative of the steps, not the actual UI. You are trying to target getting Story sequence correct first, then the number of steps in the journey, then start adding content, and lastly adding the UI layer to it What makes this process unique is 1) the number of feedback loops you can get in ( more than if you just 'design' it and user test it. 2) it builds understanding and confidence each loop of iteration in the process. Its kind of like that Computer science article "Programming as theory building" With this method, you start with a base set of understanding and theory, and at each round of feedback and iteration, you learn more, get more confident in your learnings. Based on all that, you might help me with some headlines