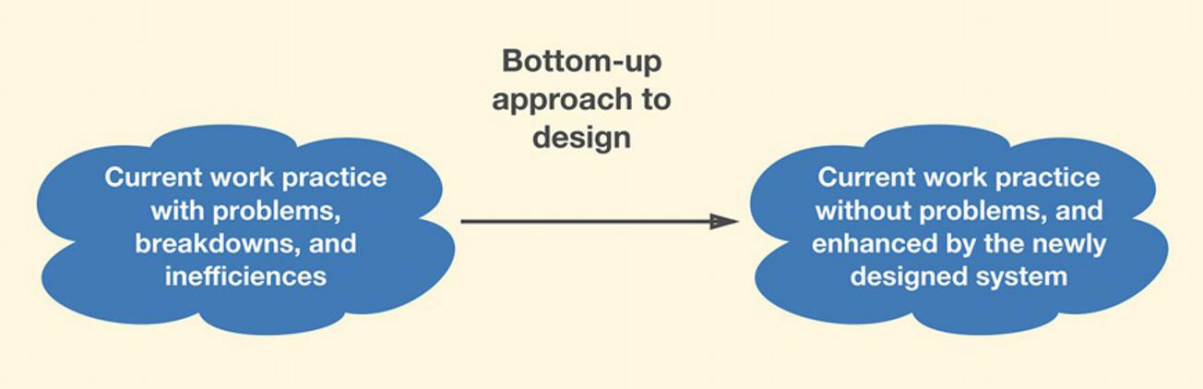
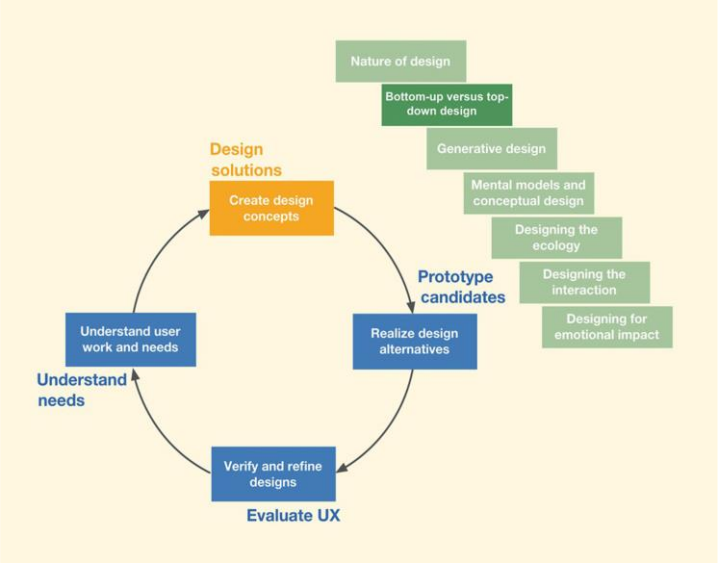
1. Explain Bottom-up Design approach

- Bottom-up design is an approach to design that starts with the details known about the work domain, work practice, and how a product or system is being used and will be used. The design is then built up in a way that will support this known usage behavior.
- This bottom-up approach is predominantly a translation exercise done through a series of transformations on a path from work activity notes to models to UX requirements to design.
- The premise is that these requirements, if met in the new design solution, will solve the problems of the users in this work practice, help users be more productive, and satisfy business mandates
- In bottom-up design, usage research is to analyze and model the existing work practice to improve it.



In a bottom-up approach, we would conduct usage research studies of voting with election booths to understand the current voting work practice in Virginia. We create work activity affinity diagrams, flow models, and other models to describe:

- Prevoting preparation:
- How citizens register to vote.
- How they find the nearest voting stations.
- What they need when they show up at the voting stations.
- The onsite configuration:
- How voting booths are set up.
- How voters interact with the booths.
- The workflow of voting:
- Voter identification.
- Checking voters off the rolls.
- Guiding voters to the appropriate booths.
- Postvoting follow-up:
- Counting ballots.
- Combining tallies from all sites.
- Announcing winners.

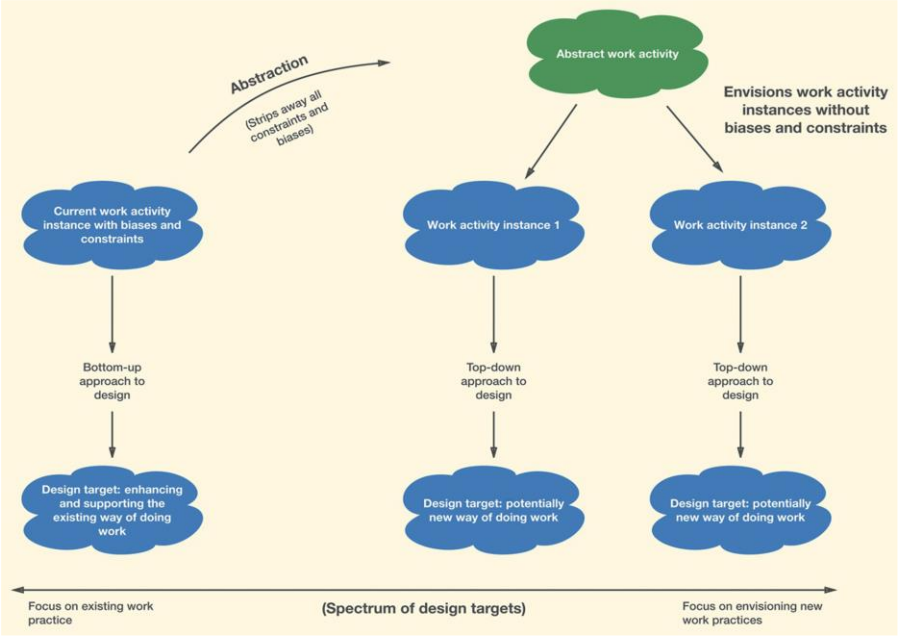


2. Explain Top-Down Design approach

- In top-down UX design creation, the goal is to create the best design solution that enhances and supports the fundamental nature of work irrespective of current practices, preferences, traditions, or constraints.
- The primary driver in the top-down approach to design is the designer and the designer's knowledge, skills, experience, and intuition.
- In top-down design, usage research is to formulate and understand the abstract conceptualization of underlying essentials of the work.

The top-down approach. Using a top-down approach based on an abstract work activity, we would:

- Focus on designing the best way to get eligible users to elect someone from a list of candidates.
- Bring into consideration all ways, including voting booths, to make this work happen.
- Consider a smartphone app or a website to vote from the comfort of the home?
- Or a 1-800 number where citizens can call and cast their votes?
- An Internet-connected device in the home that could be used for voting and that could also be used for other expressions of opinions, including petitions, surveys, and likes or dislikes of products.
- Allow for possibilities where citizens can vote over a period of time leading up to a deadline.
- Even provides for flexibility, allowing citizens to change their vote after it is cast up to the deadline—something mail-in ballots do not allow:
- Perhaps new information about a candidate was unearthed in the interim.
- Emphasize designing for a very different envisioned ecology (in fact, defining the ecology is part of the design).
- Design for the interaction (workflows, etc.), information needs (for example, full display of candidates and their party affiliations, which offices a given voter can cast ballots for, full description of each side of all issues and proposals up for a vote) and emotional needs after ecological design is completed.



Characteristics of Top-Down Design

- Top-down design can be perceived as visionary.
- Top-down design is heavily driven by domain knowledge.
- Being a potential user in the domain is good for a designer.

3. Differentiate Bottom-up and Top-Down approaches

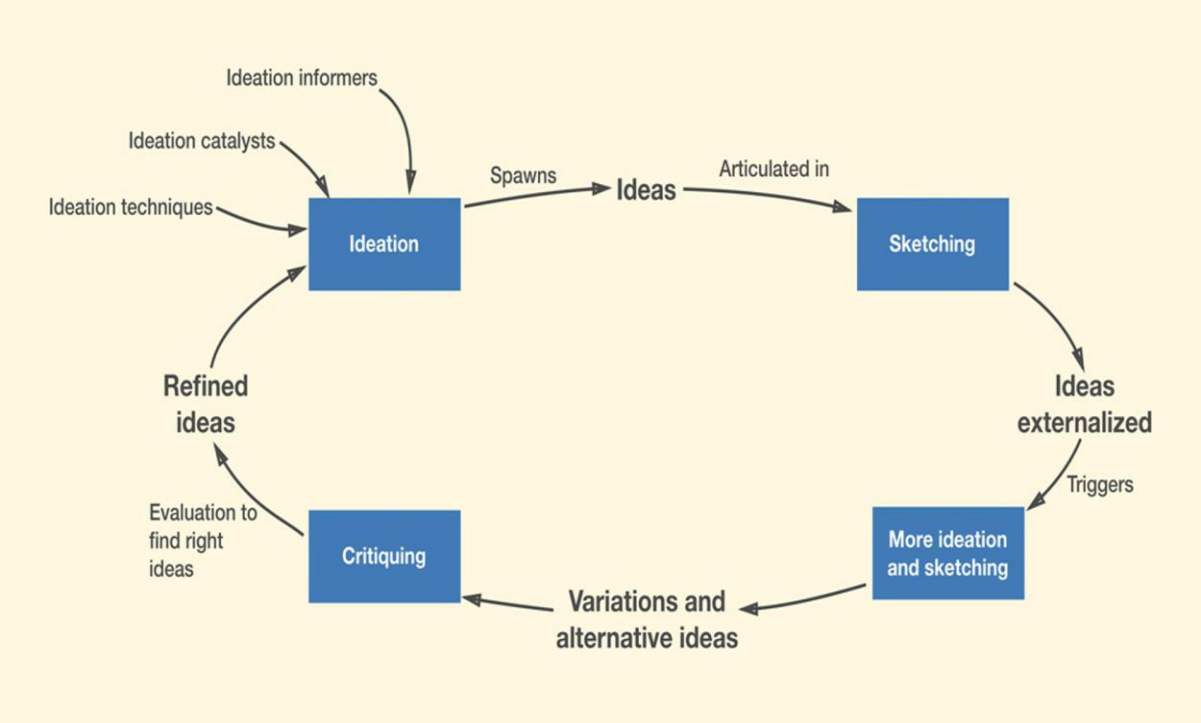
No.	Top-Down Approach	Bottom-Up Approach
1	In this approach we focus on breaking up the problem into smaller parts.	In bottom up approach, we solve smaller problems and integrate it as whole and complete the solution.
2	Mainly used by Structured Programming language such as COBOL, Fortran, c, etc.	Mainly used by Object Oriented Programming language such as C++, C#, Python.
3	Each part is programmed separately therefore contain redundancy.	Redundancy is minimized by using data encapsulation and data hiding.
4	In this the communication is less among modules.	In this module must have communication.
5	It is used in debugging, module documentation, etc.	It is basically used in testing.
6	In top down approach, decomposition takes place.	In bottom up approach composition takes place.
7	In this top function of system might be hard to identify.	In this sometimes we cannot build a program from the piece we have started.
8	In this implementation details may differ.	This is not natural for people to assemble.

4. Explain overview of Generative Design

- Designers perform generative design, an approach to design creation involving ideation, sketching, and critiquing in a tightly coupled, but not necessarily structured, iterative loop for exploring a design idea,

- The role of prototype is played by sketches, and the role of evaluation is carried out by discussion and critiquing.

- The output of generative design is a set of alternatives for conceptual designs and other capabilities or patterns for each level of the needs pyramid, mostly in the form of annotated rough sketches or storyboards.



5. List and explain Generative Design activities in brief.

Generative Design activities are listed below:

- **Ideation**: The activity where ideas are spawned. A cognitive technique to create varying and innovative design possibilities.
  Ideation is the process of creating various and innovative proposals for ecological, interaction, and emotional designs. This is a hugely creative and fun phase

- **Sketching**: An externalization activity that captures those ideas in concrete representations. Sketching is the rapid creation of freehand drawings expressing preliminary design ideas, focusing on concepts rather than details. Sketching is the rapid creation of freehand drawings expressing preliminary design ideas, focusing on concepts rather than details.

- **Critiquing**: An analysis activity to evaluate the emergent design ideas for tradeoffs

- **Refining**: An activity (usually iterative) where ideas are adopted, modified, or discarded.

6. What is Ideations?

- Ideation is the process of creating various and innovative proposals for ecological, interaction, and emotional designs. This is a hugely creative and fun phase
- Diversity in the design team helps with ideation because it brings varied perspectives.
- Ideation is the time to get clients and users to participate. The activity where ideas are spawned. A cognitive technique to create varying and innovative design possibilities.
- Ideation is the most important generative aspect of design; it is the leading edge of the act of creation. Ideation is the process of creating various and innovative proposals for ecological, interaction, and emotional designs. This is a hugely creative and fun phase.

7. What is Sketching?

- Sketching is the rapid creation of freehand drawings expressing preliminary design ideas, focusing on concepts rather than details.
- The idea of sketching as an indispensable part of design goes back at least to the middle Ages. Consider da Vinci and his famous sketchbooks. Nilsson and Ottersten (1998) describe sketching as an essential visual language for brainstorming and discussion.

Characteristics of sketching

- Everyone can sketch; you do not have to be artistic.
- Most ideas are conveyed more effectively with a sketch than with words.
- Sketches are quick and inexpensive to create; they do not inhibit early exploration.
- Sketches are disposable; there is no real investment in the sketch itself.
- Sketches are timely; they can be made just in time, done in the moment, provided when needed.
- Sketches should be plentiful; entertain a large number of ideas and make multiple sketches of each idea.
- Textual annotations play an essential support role, explaining what is going on in each part of the sketch and how.

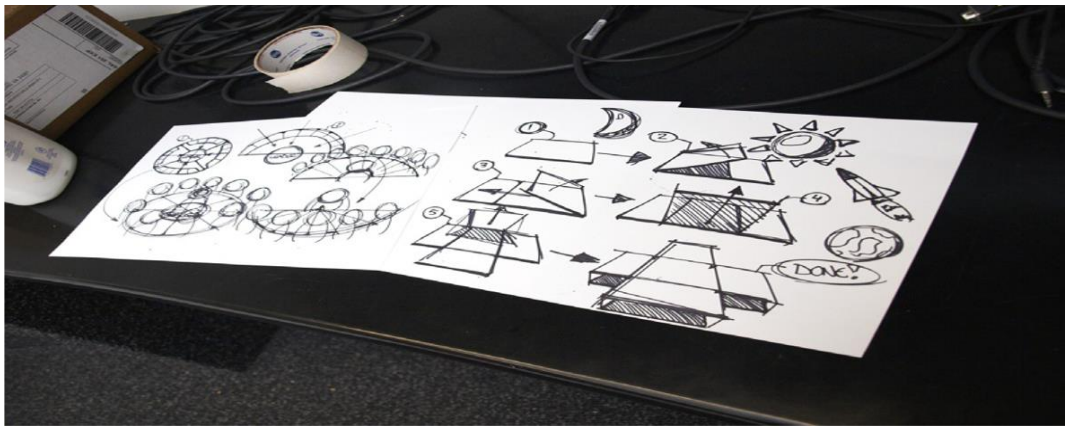


Fig. A sketch to think about design (photo courtesy of Akshay Sharma, of the Virginia Tech Department of Industrial Design).

8. List advantages of Prototyping

- Provide a platform to support UX evaluation with users.
- Offer concrete baseline for communication between users and designers.
- Provide a conversational "prop" to support communication of concepts not easily conveyed verbally.
- Allow users to "take the design for a spin" (who would buy a car without taking it for a test drive or buy a stereo system without first listening to it?).
- Give project visibility and buy-in within customer and developer organizations.
- Encourage early user participation and involvement.
- Give the impression that design is easy to change because a prototype is obviously not finished.
- Afford designers immediate observation of user performance and consequences of design decisions.
- Help sell management on an idea for new product.
- Help affect a paradigm shift from an existing system to a new system

9. Explain Depth and Breadth of prototyping with a appropriate diagram.

- The idea behind prototypes is to provide fast and easily changed early views of an envisioned UX design.
- Because it must be quickly and easily changed, a prototype is a design representation that is in some way(s) less than a full implementation.
- The choices for your approach to prototyping are about how to make it less. One way you can make it less is by focusing on just the breadth or just the depth of the system.
- When you slice the features and functionality of a system by breadth, you get a horizontal prototype.
- And when you slice by depth, you get a vertical prototype

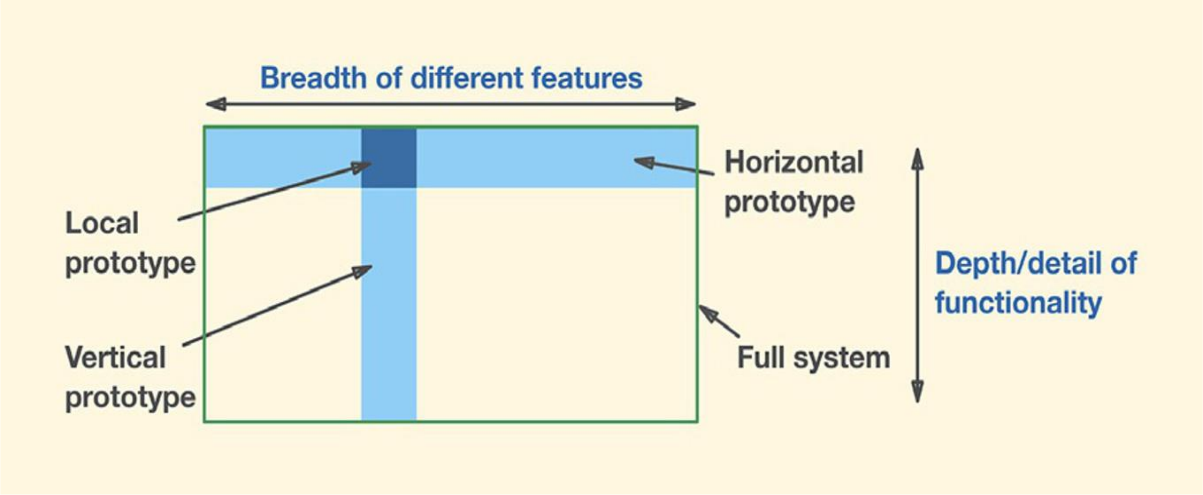


Fig: Horizontal and vertical prototyping concepts, adapted from Nielsen (1993), with permission.

Horizontal Prototypes

- A horizontal prototype is very broad in the features it incorporates, but offers less depth in its coverage of how that functionality works.
- It provides an overview on which you can base a top-down approach.
- Effective in demonstrating the product concept and for conveying an early product overview to managers, customers, and users.
- Horizontal prototypes usually do not support complete workflows, and user experience evaluation with this kind of prototype is generally less realistic.
- Prototyping in the early funnel tends to be horizontal in nature.

Vertical Prototypes

- A vertical prototype contains more depth of detail for some functionality, but only for a narrow selection of features.
- A vertical prototype allows testing a limited range of features but those functions that are included are evolved in enough detail to support realistic user experience evaluation.
- Often the functionality of a vertical prototype can include a stub for or a connection to an actual working backend database.
- Is ideal, when you need to represent completely the details of an isolated part of an individual interaction workflow in order to understand how those details play out in actual usage.
- They are most commonly used in the late funnel part of the process.

Local Prototypes

- A prototype that is narrow in both dimensions, limiting its focus to a localized interaction design issue.
- Is used to evaluate design alternatives for particular isolated interaction details, such as one dialogue box, the appearance of an icon, the wording of a message, or the behavior of an individual interaction object.
- A local prototype is the solution for those times when your design team encounters an impasse in design discussions where, after a while, there is no agreement and people are starting to repeat themselves.

10. What is T Prototyping?

- A “T” prototype combines advantages of both the horizontal and vertical prototypes, offering a good compromise for design evaluation. Much of the feature breadth is realized at a shallow level (the top of the T), but a few parts are done in depth (the vertical part of the T).
- It is recommend the T prototype because it provides a nice balance between the two extremes, giving you some advantages of each.
- Once you have established a system overview in your horizontal prototype, as a practical matter the T prototype is the next step toward achieving some depth.

- The horizontal foundation supports evolving vertical growth across the whole prototype.

11. Explain Wireframe Prototyping.

- A wireframe is a static, low-fidelity representation of your product, and in the world of web and mobile design, a basic guideline of your website or app – the skeletal framework – for both designers and developers to follow.
- For instance, developers tend to use wireframes to better understand the core functionality of a website or app, whereas designers may use them to show the navigation flow between site screens.
- A wireframe is a sketch, image, or prototype of a single interaction page or screen (in the broadest sense of "screen").
- Wireframes are described as two-dimensional sketches or drawings consisting of lines, arcs, and vertices (thus the name wireframe), plus some text for labels, representing the layout of an interaction design for a page or screen.
- These wireframes are best generated with a software tool (such as Sketch).
- Wireframe Design Elements
- Low-fidelity wireframes usually do not have graphical design elements such as images or specific colors or typography.

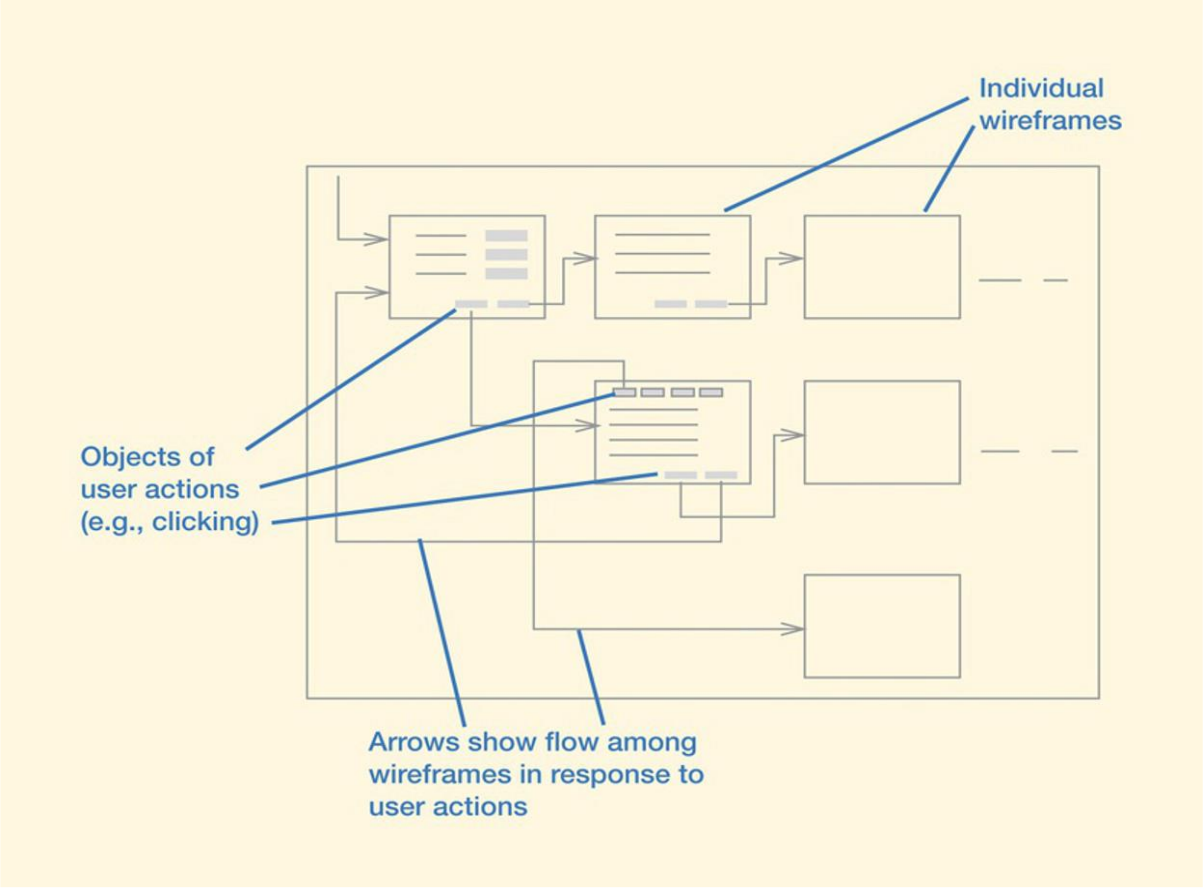


Figure: Generic Wire flow diagram

Typical elements represented in a wireframe can include:

- Header.
- Footer.
- Content areas.
- Labeling.
- Menus.
- Tabs (possibly with drop-downs).
- Buttons.
- Icons.
- Pop-ups.
- Messages.
- Navigation bar, navigation links.
- Placeholders for logo and branding images.
- Search field.

The benefits of wireframing

- Wireframes are fast, cheap and easy to create, and quick to be approved. You can therefore, properly plan the basics before moving forward, avoiding rework.
- Starting off with paper wireframes, your initial sketches will be low fidelity, low cost, and low energy.
- Wireframes are easy for stakeholders to visualize and digest from early on without getting bogged down in the details.