

< TITLES

# DESIGNING by DRAWING

OR? →

THINK, DRAW,  
DESIGN

A GUIDE TO  
CREATING

USABLE  
INTERACTIVE  
SYSTEMS

## SUBTITLE

A practical guide  
to creating usable  
interactive design

X

Colonel Hooper's -  
Field Guide To  
Interactive Design  
Principles.

Steven Hooper

# **Designing by Drawing**

A practical guide to  
creating usable  
interactive design

Steven Hooper

13 April 2009



Designing by Drawing: A practical guide to creating usable interactive design

Author: Steven Hooper

Keywords: Design, interaction, interface, specification, presentation, project, product, team, development, drawing, features, patterns, templates, grids, styles

Design by Steven Hooper

Typeset in Slimbach (which is what Robert Slimbach produced while working on Garamond, and I think has a more hand-made feel) with Akidenz Grotesk as titles and a few other bits. Oh, and the Czech and Greek characters had to be rendered in Times, but don't tell anyone.

©2009 Steven Hooper. All rights reserved.

Certain illustrations and quotes have been reproduced with permission, or are in the public domain.

No part of this book may be reproduced in any form or by any electronic or mechanical means including information storage and retrieval systems, without permission in writing from the author. The only exception is by a reviewer, who may quote short excerpts in a review.

Printed in the United States of America

Second variation

Released 13 April 2009

A few typos and title page formats revised:  
23 April 2009

intro

## How to use (and change) this document:

intro ¶1

1 I have been gathering my process documentation for ten years, and writing what could be considered a book for close to five. 2 A key problem with finishing it has been that I am not quite narcissistic enough to believe I am entirely right, so change the process all the time.

intro ¶2

1 This publication is conditional, temporary and transitory. 2 Though you can buy a copy, it is immediately out of date. 3 This will be a living document, and you can help change it yourself.

intro ¶3

1 As you read through the book, note the blue location marks, to the left of each paragraph and title, and before each sentence. 2 These indicate the section, subsection, paragraph and sentence. 3 If you think I have done something impossibly stupid, or have another way of performing a task, or just want to comment, go to <http://dbd.littlespringsdesign.com> and make your comment. 4 Note the location indicated in blue that your comment applies to. 5 Since sections may be added, and I might need to change numbering, please also reference which edition you are using.

intro ¶4

1 In the right column on some pages are blue paragraphs of text. 2 These are notes, disagreements or other contributions made by others, that have been included in the document. 3 Yours can be there also if you make a lucid, relevant contribution.

intro ¶4 s3

Your comments will appear over here. Get it?

– Clark Kent

*Planet Interactive*  
[ckent@planetint.net](mailto:ckent@planetint.net)

intro ¶5

1 Really good points, or corrections, or just my changing opinions, will be included in the body copy of future revisions, often without annotation as to their source.

intro ¶6

1 Really good contributions, and any comments where you allow us to attribute you by name will receive a free (digital only) copy of the next version released. 2 Updates will be irregular, and largely based on how many comments I get, and when I have time to reconsider and revise.

intro ¶3 s2  
 Also be aware that footnotes and end notes don't exist. Any annotation, reference or just commentary that doesn't fit in the narrative will be over here in gray type.

# Table of Contents

Prologue

## Section 1 – Theory of Design

### 1.1 What is Design?

- 1.1.1 No One Really Knows
- 1.1.2 Good Design, or Functional Design
- 1.1.3 Populist Design Today
- 1.1.4 Why is Design Scary?
- 1.1.5 Teaching How To Design
- 1.1.6 Drawing Internally, Designing Externally
- 1.1.7 Art, Expression and Realization of Design
- 1.1.8 Drawing by Dragging
- 1.1.9 The Prototyping Trap
- 1.1.10 The Home-page-fist Trap
- 1.1.11 First, Do No Harm

### 1.2 Process, Procedure, Methodology & Analogy

- 1.2.1 The Trouble with Process
- 1.2.2 Process – Making the Business Work
- 1.2.3. Procedure – Working With Others
- 1.2.4 Methodology – How You Design
- 1.2.5 Analogy – Interactive Design is Exactly Like Making a Pizza

### 1.3 Systems of Information

- 1.3.1 Making Sense of Information
- 1.3.2 Information Design?
- 1.3.3 Truth or Information?
- 1.3.4 Contextual Appropriateness
- 1.3.5 Design Responsibility

### 1.4 Information Design as a Guiding Principle

- 1.4.1 Systems of Designers
- 1.4.2 Interaction Design, Information Architecture and Everything Else
- 1.4.3 Hypermedia or Application?
- 1.4.4 Gathering, Collecting and Starting to Analyze
- 1.4.5 Collect
- 1.4.6 Systems, Processes & Integration
- 1.4.7 Benchmark
- 1.4.8 Understanding Design Through its Morphology
- 1.4.9 Creating Design Objectives

### 1.5 Divining and Prioritizing Requirements

- 1.5.1 Business Requirements
- 1.5.2 Functional Requirements
- 1.5.3 Personas, Profiles & Segments
  - 1.5.3.1 Personas
  - 1.5.3.2 Segments
  - 1.5.3.3 Profiles
- 1.5.4 Feature Evaluation
- 1.5.5 Task Inventory
  - 1.5.5.1 Use Cases and Task Lists

## 1.5.5.2 Consider the Negatives

**Section 2 – A Process for Design**

- 2.1 A Methodology for Information Design
  - 2.1.1 So, Here's the Process?
  - 2.1.2 When Do I Get to Draw?
  - 2.1.3 Summary of the Information Design Phase
  - 2.1.4 Actually, First Gather
  - 2.1.5 Logistics and Process
  - 2.1.6 Filter
  - 2.1.7 Group
  - 2.1.8 Prioritize
  - 2.1.9 Re-Filter
  - 2.1.10 Arrange
  - 2.1.11 Optimize
  - 2.1.12 Process vs. Artifact – Designing within UCD
- 2.2 Sketching Towards Understanding
  - 2.2.1 Box Models
  - 2.2.2 Lists
  - 2.2.3 Post-Its
  - 2.2.4 Blue box
  - 2.2.5 Whiteboard
- 2.3 Formal Modeling Methods
  - 2.3.1 Modeling Languages
  - 2.3.2 States & Time
    - 2.3.2.1 Drawing States
    - 2.3.2.2 Drawing Time
  - 2.3.3 A Unified Modeling Language
    - 2.3.3.1 Interaction Diagramming
    - 2.3.3.2 UML, More or Less
    - 2.3.3.3 Explicit Depictions of Time
  - 2.3.4 Wireframes Grids & Templates
    - 2.3.4.1 Grids
    - 2.3.4.2 Wireframes
      - 2.3.4.2.1 History & Background
      - 2.3.4.2.2 Enough Detail, But Not Too Much
      - 2.3.4.2.3 Moving From Drawing to Drawing
    - 2.3.4.3 Templates
    - 2.3.4.4 Object Oriented Design

**Section 3 – Artifacts**

- 3.1 Drawings are Objects
- 3.2 Using Artifacts to Design
- 3.3 Concept of Design
  - 3.3.1 Creating and Presenting Design Options
  - 3.3.2 Creating Concept Documents
  - 3.3.3 Synching and Revising
- 3.4 High Level Interaction Designs

- 3.4.1 From Sketching to Specifying
  - 3.4.1.1 More Box Models
  - 3.4.1.2 Moving to Computers
  - 3.4.1.3 Process, Flow and Interaction
  - 3.4.1.4 Details and Specifics
  - 3.4.1.5 Concept Documentation
- 3.4.2 Numbering and Annotation
  - 3.4.2.1 Breaking the Numbering
- 3.4.3 Technology of Drawing
  - 3.4.3.1 Versioning
- 3.4.4 Types of Documents
  - 3.4.4.1 Other Drawing Solutions
  - 3.4.4.2 Distribution and Delivery
- 3.5 Detailed Interface Designs
  - 3.5.1 Naming and Understanding
  - 3.5.2 Specifying and Drawing
    - 3.5.2.1 Details and Documents
    - 3.5.2.2 From Words to Boxes to Pictures
  - 3.5.3 Content, Brand and Style
  - 3.5.4 Technologies for Drawing
    - 3.5.4.1 Scale
    - 3.5.4.2 Type, Color and Graphics
    - 3.5.4.2 Templates and Guides
  - 3.5.5 Documentation
    - 3.5.5.1 Big Wireframes
    - 3.5.5.2 Annotated Wireframes
    - 3.5.5.3 Multi-page Wireframes
    - 3.5.5.4 Specification Documents
    - 3.5.5.5 Object Oriented Documentation
    - 3.5.5.6 Distribution
- 3.6 Style Guides & Graphics Deliverables
  - 3.6.1 Style Guides
  - 3.6.2 Production Graphics
- 3.7 Prototyping & Presentation Layer Deliverables
  - 3.7.1 Prototypes
  - 3.7.2 Code
- 3.8 Delivering and Presenting Artifacts
  - 3.8.1 Labeling
  - 3.8.2 Presentation Tips
  - 3.8.3 Procedures
- 3.9 Iterating on Detailed Design
  - 3.9.1 Iterating on Design
  - 3.9.2 The Project Lifecycle

## Prologue

pro

pro ¶1

1France Rupert, then a co-worker, asked me one day several years ago what sort of book I'd write if I had the chance to, and I surprised him – a lot – but just showing him this book as a fifty page Word document. 2Then he asked me a good question, what is it about? 3Or, more importantly, what makes it different from any other interaction design or usability book?

pro ¶2

1And... I didn't have an answer. 2I had created a book-length collection of design processes and ideas without a single unifying theme. 3So I worked on it, and changed it, and after a few more years of irregular work, I think came up with a single narrative thread.

pro ¶3

1Now my answer is two things. 2First of all, it's about designing interactions. 3About designing. 4Not just a list of good designs, not just the input and outputs, but literally how to design.

pro ¶4

1“And-then-a-miracle-happens” seems to be the assumed process for the designer. 2This might be fine if we could always hire brilliant designers who throw out perfect ideas their first time. 3But as the field grows, and technology changes every month, it is increasingly important that non-designers be able to perform some of these same tasks, or at least be able to communicate with the design process.

pro ¶5

1Understanding your own process also helps you do your job better. 2While I can't show you all the math here as I forgot to save all this information, and some of it is still proprietary, I have determined this to be true by implementing process and analyzing the results. 3You'll just have to take my word for it for now.

pro ¶6

1And secondly, it's about artifact creation. 2A lot of design theories hold that artifact creation is inherently bad; the mere process of moving from your head to paper sullies the perfection of the idea. 3In the past decade this movement has emerged into a strong trend to take any concept, scribble an idea on a whiteboard, and move straight to Flash or some other prototyping tool. 4Without getting too ahead of ourselves, that's a few steps too far, and hinders the design process

in subtle, insidious ways.

- pro ¶7 1By artifacts I mean documents that are consumed by others. 2Things that you share with the business owners, or pass on to development to get things made.
- pro ¶8 1While I have not tried to teach drawing skills, I refer here to drawing a lot. 2It may or may not be pencil and paper, but it's never "sketching" by prototyping. 3Drawing by putting lines down on a page is key to the design process, and both the closest thing to dumping your brain on paper, and as I hope to show, a way to improve your design by the very act of setting it down.
- pro ¶9 1I would avoid using this as a textbook or workbook, at least until you have read through it completely without doing homework, at least once. 2Mostly, because I am not that good at organizing, so some answers are strewn throughout the book; if you don't read it all you won't get the whole story and some of these processes or procedures will seem incomplete, impossible or stupid.

# Section 1

# **Theory of Design**

## 1.1 What is Design?

### 1.1.1 No One Really Knows

1.1.1 ¶1 1Well, actually, there are any number of people who could give a valid argument as to what design is today, what it was yesterday and even what it might be tomorrow.

1.1.1 ¶2 1I, however, am not one of them. 2We're only interested in it from a relatively functional point of view. 3If you want to talk about the philosophy, there are any number of books entirely consumed with the task (some are in the bibliography if you are interested).

### 1.1.2 Good Design, or Functional Design?

1.1.2 ¶1 1So, a good start to determining what design is might be to determine what it means to the designer, from a functional, effective, end-product point of view.

1.1.2 ¶2 1 Kauffman's twelve precepts of Good Design (1950, MoMA) are informative to review for designers even today:

1. Fulfill the practical needs of modern life
2. Express the spirit of our times
3. Benefit by contemporary advances in the fine arts and pure sciences
4. Take advantage of new materials and techniques and develop familiar ones
5. Develop the forms, textures and colours that spring from the direct fulfilment of requirements in appropriate materials and techniques
6. Express the purpose of an object, never making it seem to be what it is not
7. Express the qualities and beauties of the materials used, never making the materials seem to be what they are not
8. Express the methods used to make an object, not disguising mass production as handcraft or simulating a technique not used
9. Blend the expression of utility, materials and process into a visually satisfactory whole
10. It should be simple – its structure evident in

1.1.2 ¶2  
Edgar Kaufmann Jr., Director of Industrial Design at MOMA in New York spoke out against a 'style follows sales' mentality by putting on a series of Good Design exhibitions at the museum between 1950 and 1955. This emphasized the European Modernist aesthetic and built upon the design tendencies that had been apparent since the establishment of the Department of Architecture and Industrial Art at the Museum in 1932.

its appearance, avoiding extraneous enrichment

11. Master the machine for the service of people
12. Serve as wide a public as possible, considering modest needs and limited costs no less challenging than the requirements of pomp and luxury

### 1.1.2 ¶3

A reflection on the Good Design ethic, and its place in modern craft, design and culture occupies George Marcus' book *What is Design Today*, 2002, Harry N. Abrams

### 1.1.2 ¶3

1A lot of these still make a lot of sense, — furniture designers working today put these to good use in their day-to-day work — but they also imply a specific style of design. 2Of course, this was intentional at the time, and guides a specific style of design in furniture, interior design and decoration.

### 1.1.2 ¶4

1For interaction design, I am personally unconcerned with additional adornment, or the style of the design, as long as it does not interfere with the primary functionality. 2There's room to argue what involves adornment that interferes, hence the original Good Design movement, and many others.

### 1.1.2 ¶5

1But much of the concepts, of both design and that designed products are mass-distributed (by factories originally, now by that as well as time-sharing on computer networks) I might modify these precepts, therefore, as follows for contemporary interactive design purposes:

1. Fulfill the practical needs of your users
2. Express the spirit of our times
3. Benefit by contemporary advances in technology and the understandings of human behaviors
4. Take advantage of new technologies and techniques and develop new ones
5. Allow the form, and position of each element to spring naturally from the direct fulfillment of requirements, using appropriate techniques and methods
6. Express the purpose of an element, never making it seem to be what it is not
7. Express the qualities and truth of the information presented, never making information seem to be what it is not
8. Emphasize transparency in process, intent and information processing, to gain the faith

- and trust of your users
- 9. Blend the expression of utility, technology and process into a visually satisfactory whole
- 10. It should be simple – its meaning and content evident by its appearance, avoiding extraneous enrichment
- 11. The system should work for the end user, instead of for its own means
- 12. Serve as wide a public as possible, considering modest or specialized needs, and limited capabilities as no less challenging than the requirements of your most expected or profitable users

1.1.2 ¶6 1These are still pretty esoteric-sounding, so what does that mean? 2Well, basically, that functionality should never be degraded, obscured or damaged by the design of the information.

1.1.3  
Populism in social or political discourse, at least today, is not exactly applicable to the understanding I have always had of populist design. I might mean something more like “egalitarian design,” where every user’s opinion is valid and important to meet.

1.1.2 ¶7 1I’ll have a lot more to say about information, presentation and design later.

1.1.3 **Populist Design Today**

1.1.3 ¶1 1The original Modern Design movements ended up with a few good results (e.g. Isamu Noguchi’s Akari lamps), but mostly by creating high-end, relatively unachievable items (practically anything by the Eameses).

1.1.3 ¶2 1A few modern retailers, like IKEA and Target, are becoming relatively successful at offering good design (if not Good Design) to the masses. 2Sometimes this works because the populace is aware of the design ethic on some level, and sometimes it’s just is a side effect of convenience and price.

1.1.3 ¶3 1However, interactive systems, especially as a result of internet technologies, present an unprecedented opportunity to offer design to the people, not to mention of and for them. 2Interactive systems are used by many people, often very frequently, in very personal ways.

1.1.3 ¶4 1Much interactive design is terrible, of course, but good design tends to pay off, and simple, easy-to-use systems tend to attract more users than complex, difficult interfaces.

1.1.3 ¶3  
You should talk about the five different kinds of design (Accidental, et. al.) that you referred to before -- from Spool’s article at [http://www.uie.com/articles/five\\_design\\_decision\\_styles](http://www.uie.com/articles/five_design_decision_styles) – Christopher Nemeth Designer

Good point. I’ll add that in next time, since I blogged about it even: <http://www.littlesprings-design.com/blog/blog/2009/01/23/style-procedure-and-success/> – SH

1.1.3 ¶5 1The real key is the opportunity for change. 2A chair is going to remain broadly unchanged throughout its lifecycle, and this may last for years. 3Partly, this is the result of manufacturing processes; tooling costs and other setup functions of a manufacturing concern are often an all but insurmountable portion of the cost of making. 4Change only occurs when needed (for competition) and often only at a surface level to save on these costs.

1.1.3 ¶6 1Purely software-based interactions offer an opportunity for interactive systems to be improved on a regular or even ongoing basis. 2This is most true of systems like websites, where the bulk of the interaction can vary with each visit or even page, but updates and upgrades over the network allow similar flexibility for installed software.

1.1.3 ¶7 1Armin Vit has asked where are all the landmark web designs? 2The question might seem useful from the classic point of view of art & design criticism, education and industry understanding. 3But I believe it's a flawed concept.

1.1.3 ¶8 1Great concepts, solutions or (dare I say it?) design patterns can add to the general understanding of what is a good design, and be used by others. 2And as much as I like large iron things, and beautiful, printed matter, the concept that there is permanent, unchanging, sort of timeless interactive product does not seem valid.

1.1.3 ¶9 1Not because of the design downsides of network-based updating, or the rapidly cycling business model for software, but by empowering your users via satisfying their changing demands.

### 1.1.4 Why is Design Scary?

1.1.4 ¶1 1This is basically why I started typing out my thoughts and processes of design. 2Not just because I am narcissistic, but because people I work with kept needing to be designers.

1.1.4 ¶2 1Design was troubling to some of these people. 2Disturbingly so. 3I saw people avoid it, practically shirk it. 4Good, smart people. 5Often, much smarter and more generally capable than

1.1.3 ¶7  
Is design really an “open source” concept?  
– Christopher Nemeth  
Designer

1.1.4  
I find incorporating drawing early in a process an affective way to disarm naysayers. Since visual art is quite subjective I often found that when I drew in a vacuum and presented my work at the end of a long work session in my cube, I got quite a few team members that questioned my design solutions. They often asked “did you think of this or that.” Often I did think of their points, they were just unaware of my thought processes. When I started actively drawing in meetings, carrying my sketch pad, setting up brainstorming sections and drawing on whiteboards, I found that at the end of design iteration I had more buy in to my design. It helped my team members have faith in my ability to do good design.  
– Leslie Salazar-Bushell  
Interaction Designer  
Pearson eCollege

1.1.3 ¶7  
Armin Vit asked this on his blog in a post “Landmark Web Sites, Where Art Thou?” [www.underconsideration.com/speakup/archives/004033.html](http://www.underconsideration.com/speakup/archives/004033.html)

## 1.1.4 ¶4

The definition of design “The act of working out...” is repeated so many places, including numerous dictionaries, I cannot figure out where it came from. Seems pretty universal to me.

- me. 6So, why are they scared of design?
- 1.1.4 ¶3 1Well, lets start with semantics. 2It is probably worth looking at what design means, at least a little bit. 3What does the word “design” actually mean? 4Here’s one definition that suits me right now:
- 1.1.4 ¶4 *1The act of working out the form of something, as by making a sketch, outline or plan.*
- 1.1.4 ¶5 1So design means you need to draw to express the idea. 2Well, a plan would do, but I think they mean like a blueprint, still, not a schedule of times or anything.
- 1.1.4 ¶6 1So, it’s about drawing. 2Drawing is indeed scary. 3Lots of people think they don’t have a clue how to draw. 4Grown men freeze up at a blank piece of paper. 5Go to a design conference and watch a hands-on session. 6Make bets on which of the five designers in a group gets “stuck” with drawing the results of the exercise (hint: its usually me). 7If half of these people draw for a living, how can anyone else be expected to draw?
- 1.1.4 ¶7 1Well, blame lack of art funding in grade school if you want. 2Do you remember drawing as a youth? 3Probably. 4Why? 5Because visual communication is a powerful medium, and that memory of your early ability to do so is wired in like that. 6All you need to do is grab onto that drawing sense again, and try it.
- 1.1.4 ¶8 1Trying it is easy. 2Start drawing box diagrams when in meetings. 3Stand up and contribute to the whiteboard sketching when it comes time to design. 4Don’t be worried about your ability to draw a straight line, or that you stick to one color, and feel free to tell people like me who mock your skills (like me) to shove it.
- 1.1.4 ¶9 1Once you do this enough, you’ll become habituated — again, like when you were young — to using drawing as a way to express your ideas. 2Absorbing this philosophy of design can lead you to doing it automatically. 3You can start to think, analyze and design as a routine part of your work and life.

## 1.1.4 ¶2

People expect design to elicit an emotional response, like art traditionally has.

– Christopher Nemeth  
Designer

## 1.1.5 Teaching How to Design

1.1.5 ¶1 1One problem cropped up almost immediately. How do I design? 2This is not something taught in art (or design) school. 3You are certainly taught what I'll call "what" to design, but not exactly how. 4There's all sorts of information on composition and color and perception and meaning and technique, but practically nothing on process.

1.1.5 ¶2 1Which is fine for art and design schools. 2Presumably people going there have a modicum of talent, or have worked out some of this basic stuff on their own or in primary school. 3But what about those who have no background or, even, interest?

1.1.5 ¶3 1I started very simply looking at what I did. 2Why I made any series of decisions, and how they built on each other. 3That, and a lot of reading led to a process. 4Which was terrible, and wrong. 5But over time (mostly, one four-year period) I formalized a series of processes and procedures for design and artifact creation.

1.1.5 ¶4 1These I then shared with my teams, and encouraged (or when I had employees, required) them to use. 2The results snuck up on me, due to the slow development of the process but over time I noticed it was working. 3Error rates were reduced, timelines were shortened. 4Technical teams, and process-centric technology organizations began to understand we were operating in a consistent, repeatable, traceable manner. 5Business teams began to understand that we were making decisions based on knowledge (and even information and science) and trusted us, over intuition, aesthetic sense and whatever the competition is doing. 6Not always, but enough to make a difference.

1.1.5 ¶5 1I certainly cannot make people be better at free form drawing. 2Maybe art school would help, but I am not that good a teacher. 3However, this is all about just drawing enough to develop and express the interaction of things like web pages or mobile phone applications; you are just making boxes and lines and lists.

1.1.5 ¶6 1Most of all, don't be self-conscious. 2Pick up a pencil, pen, crayon or marker and start drawing.

1.1.5 ¶1  
Design principles need to be absorbed into processes.  
– Christopher Nemeth  
Designer

1.1.5 ¶5 s1  
But principles of good design can be expressed, shared, learned and implemented.  
– Christopher Nemeth  
Designer

1.1.5 ¶2  
If you want to learn to draw, or need to suggest something for others who cannot spend several years being depressed in art school, try *Rapid Viz : A New Method for the Rapid Visualization of Ideas*, by Kurt Hanks and Larry Belliston. I'd get the first or second edition, used if need be. Messing with success never works and the newer one is notably worse than the older ones.

## 1.1.6 Drawing Internally, Designing Externally

- 1.1.6 ¶1 1Stop and think about the title. 2I'll put in a paragraph break so you can look up there, read it, absorb it and come back to the beginning of the second graph. 3Now go.
- 1.1.6 ¶2 1Does that make any sense? 2Nope. 3It doesn't. 4Not at first at least. 5I'll bet you cannot do it now. Let me explain.
- 1.1.6 ¶3 1Drawing is a visceral. 2Seriously. 3Not just the kind of metaphorical manner of something you feel in your gut, but something physical, expressed by your bones and muscles and everything else attached to the hand which moves the pencil (stylus, mouse, whiteboard marker, etc).
- 1.1.6 ¶4 1Design is, theoretically, totally ephemeral; an internalized process. 2How do you get that design out into the world? 3Picture... anything. 4Here's one: A lamp that is powered by an integral wind generator. 5Yes, this is a stupid idea, but that just means it probably doesn't exist. 6Therefore you'll have to design it. 7Okay, got it in your head? 8That's designed.
- 1.1.6 ¶5 1Now, get it out on paper. 2You need to express the design viscerally. 3By drawing. 4So, if you can think — internally — in the way you can draw, you can lay that design in your head on paper that much more easily. 5You won't get hung up on trying to express a design you cannot draw.
- 1.1.6 ¶6 1So, how to do this? 2Well, if I could tell you in a single paragraph of a single chapter, there'd be no point for the rest this writing. 3You have to read on, and try it, and practice it. 4But, you also have to know what you are doing; you have to be self-aware to make this work. 5So, stick a post-it on this page that hangs out, and look at the title occasionally, and think about it.

## 1.1.7 Art, Expression and Realization

- 1.1.7 ¶1 1Lets stick with this a minute. 2It's something a lot of smart people with their own books and lecture fees have told me I am wrong about. 3They say, design is all internalized.

### 1.1.7

I would say that art is the expression of an emotion or experience through a medium. Be it music, art, paint, or drawing. Design is a process of problem solving that is visualized. Like mathematician who continually works out an equation to it's simplest form, a designer works out a solution by continually reworking his initial drawing.

– Leslie Salazar-Bushell  
Interaction Designer  
Pearson eCollege

## 1.1.7 ¶3

I cannot, right now, find the Eric Gill quote. I swear I found it in Sebastian Carter's 1987 (updated '95) *Twentieth Century Type Designers*. This is a great book in general; even if you are only so interested in type it discusses history, the impact of technology on design and art, political and economic considerations on design and process, and the interaction (and overlap) between designers and craftsmen, long before software development offered the same conflicts.

- 1.1.7 ¶2 1All. 2Exclusively. 3Such that any externalization of it is an abstraction that suffers a generational loss. 4Like photocopying a photo, you get the impression, but it gets worse. 5I totally disagree.
- 1.1.7 ¶3 1Eric Gill once expressed part of his love of type design as that he was able to design “things, not drawings of things.” 2I claim that everything is a thing; in what manner is a drawing, even one of or intended to become a physical object, not a real thing itself? 3I feel this mentality takes away from the value of drawing and design, and thereby from the ability (or maybe willingness) of people to express themselves.
- 1.1.7 ¶4 1For starters, externalization is the only way to share. 2For the most part, painting and poetry and dance seem to be perfectly accepted as expressing the artist's vision. 3Are these less expressive because they are real?
- 1.1.7 ¶5 1For design, you can stick it on paper, and its permanent, while you change. 2Put an idea down, come back in 10-minutes and you can evaluate it yourself with a different mindset. 3Try that in a day, a week or a year. 4How does that design look? 5If you left it internalized, its always fluid, and there's no way to evaluate it.
- 1.1.7 ¶6 1There's also no way to understand what you have modified. 2No one said you can't change a design once its out on paper, and in fact its time to say “you should.” 3There are lots of specific things that should be done to get a good interactive system, and we'll be talking about those later on of course. 4But even art (visual, literary or performing) undergoes a process of evolution as a result of it actually being put on paper or performed live.
- 1.1.7 ¶7 1Note I said evolution, implying the changes trend towards the positive. 2The final design is not going to be what you came up with in your brain, but its still a product of your brain.
- 1.1.7 ¶8 *An Expressionist wishes, above all, to express himself....[An Expressionist rejects] immediate perception and builds on more complex psychic structures....Impressions and mental images that pass through mental peoples soul as through a filter which rids them of all substantial accretions to produce their clear essence [...and] are assim-*

1.1.7 ¶5 s5  
...or critique it, or evolve it, or produce it as a real thing.  
– Christopher Nemeth Designer

1.1.7 ¶7 s2  
An idea is not a design, it is merely where a design starts.  
– Christopher Nemeth Designer

*lated and condense into more general forms, into types, which he transcribes through simple short-hand formulae and symbols.”*

— Antonín Matějček

1.1.7 ¶8

The quote from Antonín Matějček is from Gordon, Donald E. (1987). *Expressionism: Art and Ideas*, Yale University Press.

- 1.1.7 ¶9 1Expressionism in visual arts is a manner of modifying conventional, strict representation in an attempt to denote emotion. 2A design of an interactive system may want to assign some value other than emotion. 3Perhaps meaning, understanding, trust, or just information. 4Regardless, your brain is collaborating with the media in which you are working, thru a process, to get to a simple, clear solution you can share and others can understand.

## 1.1.8 Drawing by Dragging

- 1.1.8 ¶1 1How can you be expressive when drawing? 2Well, one way is to draw.
- 1.1.8 ¶2 *A box of crayons and a big sheet of paper provides a more expressive medium for kids than computerized paint programs.*  
— Clifford Stoll
- 1.1.8 ¶3 1Not to beat history and etymology into the ground to make a point, but... well, actually I am gonna do it a little. 2Drawing means dragging a writing item across a page. 3Drawing in the sense of pulling things.
- 1.1.8 ¶4 1This is important. 2Feel this when you next draw. 3The point is that you cannot do this on the computer. 4You can surely draw on computers, and I do all the time. 5In fact, it's probably best to make your final deliverables on a computer. 6That way all the lines are straight and the words are readable and the drawing is infinitely reproducible and distributable. 7In fact, there are whole chapters later about suggested detailed artifact creation on computers.
- 1.1.8 ¶5 1But when starting out, just grab a piece of paper and a pencil, or marker. 2Or pick up a white-board marker. 3Draw boxes. 4Draw lines. 5Write words no one else can understand. 6Erase things with your finger, and if you can't, just scribble them out. 7Build, rebuild, deconstruct, destroy and generally, let the drawing evolve while you directly interact with it.

1.1.8 ¶2

Clifford Stoll's well-known quote is from his 1995 book *Silicon Snake Oil: Second Thoughts on the Information Highway*.

1.1.8 ¶6 1I use, almost exclusively, pen tablets for my computers. 2I have won awards for digital painting. 3It's still not the same thing. 4Get away from the computer occasionally.

### 1.1.9 The Prototyping Trap

1.1.9 ¶1 1Let's go back for a minute to the argument most design gurus make, that all design is internal, and any production dilutes it. 2This has, over the last few years especially, been extended to mean that you should go as rapidly as possible from your brain to prototyping. 3In the edges is the ability to go straight to final, production-grade software products.

1.1.9 ¶2 1None of that pesky design, so great, right? Wrong. 2And I know it's wrong because it's not new. 3Programmer/designers have been around for a long time. 4In fact I've built things directly in code without a sketch. 5I've seen this process fail in two distinct ways:

- 1 1Design processes are good. 2This whole book is about that, so I am not going to get too into it here. 3In short, going to straight to prototype discourages design explorations, revisions, even (at least for all systems I have seen) collaboration. 4It also encourages everyone looking at it, no matter how rough the presentation is made to look, to believe it is real. 5This discourages even more exploration, or if it is disliked tends to require complete redesign.
- 2 1The selected prototyping technology always adds it's own constraints. 2The problem with this is also two-fold, but I'll try not to start a new bullet list. 3The simplest issue is that the constraints of the prototyping environment will never precisely match that of the final product. 4Even if the real software can behave like the prototype, it often is not optimized for that system, or ignores interesting features of the environment.

1.1.9 ¶3 1But more importantly it restricts designers from pushing their boundaries. 2The easier a tool is to use, or the more familiar with an environment a developer gets, the more they will tend to re-use

existing code and other presentational tricks.

3But your job as is to find the best possible design for the product, not the best design from a limited subset. 4Push the boundaries.

1.1.9 ¶4 1A key argument in favor of immediate prototyping is that designers drawing on paper do not understand systems, processes and technology. 2My anecdotal answer is: look at me. 3I draw entirely (and never prototype designs myself, though I have had other do it for me later in design, sure) but am very much a go-to guy for security models, database integration and other technologies. 4I have written pseudocode and fixed UI issues in production JSP files directly in IDEs.

1.1.9 ¶5 1“How” is something anyone can achieve: go ask someone. 2You will eventually learn all this, but the first step I took and one I still use on every project is to ask the developers. 3Ask them what they develop in, and what process they use, then search the internet about it. 4Ask them to review your initial concepts, and not just to approve but to provide feedback on how they expect to implement new and weird features. 5Try to sit in on their design meetings; yes, software design is a design discipline, and you’d be surprised how much value the UI designer can have in them.

### 1.1.10 The Home-page-first Trap

1.1.10 ¶1 1There’s one part of my design process that is so core to my being I often forget to tell anyone. 2In fact, it’s so key that I have yet to find a way to include it cleanly in the processes that follow. It’s baked into them too deeply.

1.1.10 ¶2 1Book jacket design, and even the graphic design principles of book or magazine layout, are interesting, important, respectable fields. 2But essentially no one has ever started a book by designing the dust jacket. 3Content, and the intent of the content, drive design.

1.1.10 ¶3 1Similarly, for websites, never design the home page first. 2For mobile phones, never design the idle screen first. 3And so on. 4The same goes for physical product, or laying out your living room

1.1.10  
I would say that before you can design the box you have to understand what is in the box. In interaction design that is the content that is being delivered and tasks that the user has to complete.

– Leslie Salazar-Bushell  
*Interaction Designer*  
*Pearson eCollege*

furniture. <sup>5</sup>This is not to say form over function, but that content and function informs style and presentation.

1.1.10 ¶4 <sup>1</sup>This is not something that emerged for me as an interactive design principle, or even something I invented. <sup>2</sup>It came from print design, and much more from actually working as a print designer than from any education in the process of design I was taught.

1.1.10 ¶5 <sup>1</sup>Okay, sometimes you will get a client who insists they see the home page. <sup>2</sup>There are lots of client-relations strategies you can go over to try to communicate all of this to them, but in the end, even when I first have to show off the cover of a brochure, or the home page of a website, or the idle screen for an application, I'll do it. <sup>3</sup>Sort of. <sup>4</sup>I'll still have an idea, and usually pages of reasonably serious drawings of the rest of the application, even if I have to keep them to myself.

1.1.10 ¶6 <sup>1</sup>I am generally suspicious of any designer who starts with the home page, and have yet to hear good argument for why it's a good idea.

### 1.1.11 **First, Do No Harm**

1.1.11 ¶1 <sup>1</sup>Sometimes, the almost trite slogans are actually spot on. <sup>2</sup>While not part of the Hippocratic oath, a key principle for much of medicine is *Primum non nocere*, generally but slightly oddly translated as "do no harm."

1.1.11 ¶2 <sup>1</sup>This is a reminder, in medicine, to consider what any interventive therapy might do. <sup>2</sup>Not just to the immediate symptom or condition, but to others, to other treatments, and to the patient as a whole.

1.1.11 ¶3 <sup>1</sup>Interactive systems are always designed for use by large numbers of people. <sup>2</sup>A common downfall of any number of designed objects, interfaces or systems is to appeal to the average or typical user, without consideration of the whole population. <sup>3</sup>Most features are not universally beloved, and some that are liked by some are reviled by others. <sup>4</sup>It's a balancing act to not harm the perception of a product or service when improving in some other way or for some other group of users.

1.1.11

The history of this phrase “do no harm” is interesting, and possibly revealing, from the Wikipedia entry: “The origin of the phrase is not widely known; contrary to popular belief, the phrase is not in the Hippocratic Oath. However, it is often described as a Latin paraphrase by Galen of a Hippocratic aphorism... The closest approximation to the phrase that can be found in the Hippocratic Corpus is “to help, or at least to do no harm,” taken from *Epidemics*, Bk. I, Sect. V.” Other translations and different ways of presenting the concept are also worth discussing.

1.1.11 ¶4

1 While you certainly can't please all the people, all the time, do try not to actually anger too many of them. 2 Some of the analytical tools below specifically are designed to address this, but it also must be one of the over-riding principles that you keep in mind throughout the process.

## 1.2 Process, Procedure, Methodology & Analogy

1.2 ¶1 1Process is not the world saver that executives or the process consultants always say, nor as bad as it's made out to be by most of the line workers I talk to.

1.2 ¶2 1Again, semantic distinctions can be useful here. Process is applied a little too broadly. 2To build things and work with others, procedures and methodologies are important to understand as well.

### 1.2.1 The Trouble With Process

1.2.1 ¶1 1The worst of the semantic issues is a misunderstanding of iterative design (or development). 2Iterative design is great. 3It's fluid. 4It gives you a chance to fix things that you find are broken, and tweak things as you get better solutions to them. 5You can make changes based on test and research, so they are grounded in reality. 6And that change doesn't even need to violate good object-oriented principles that break the work already done.

1.2.1 ¶2 1The problem is that its almost always actually pursued as incremental design. 2Incremental design breaks up individual design elements into bits, and implements a number of these bits or functions (within IT software process, usually based on estimates of hours available) in each of several design cycles. 3Any one function or widget is usually completed within its increment. 4There is pretty much no way to change any one element later on, which is almost the opposite of iterative design.

1.2.1 ¶3 1I am not sure where this came from, but its self-sustaining now because exactly no one seems to have done actual iterative design. 2No one seems to know the distinction so it keeps going.

### 1.2.2 Process – Making the Business Work

1.2.2 ¶1 1Now that I've gone on for a while about what process is not, its time to get back to what it

#### 1.2.1

I would point out that iterative design allows for and even requires errors and issues. In iterative design a team makes a decision that a certain design will work to start out on, with the understanding that issues and errors will come out during testing. Testing will spawn rework and re-design, but in smaller amounts or even tweaks. Till the solution that works for users is developed.  
– Leslie Salazar-Bushell  
Interaction Designer  
Pearson eCollege

is. **2**Process is about business practices. **3**How to get business, how to account for time spent, how to assign people (you may call this “traffic management”) and so on.

**1.2.2 ¶2** **1**As you can see, some of these are actually performed by the design team. **2**You are likely to have internal staffing processes, for example. **3**However, since they are about internal functions, they are business-oriented, and are process.

**1.2.2 ¶3** **1**Determining if a task is a process is easy. **2**If a strict set of rules can be applied, but the result will change over time, or as internal conditions change, its a process. **3**Think of scheduling new work; any designer (ideally) should be able to do the job as well as any other. **4**Which one is assigned is dependent entirely on workload, scheduling, and generally internal and transient conditions. **5**The conditions of the designed procedure have little or nothing to do with it.

**1.2.2 ¶4**

Diving into a detailed description of incremental and iterative design is beyond the scope of the book, but worth looking up. Briefly, from the Wikipedia entry on the both: “Incremental development is a scheduling and staging strategy, in which the various parts of the system are developed at different times or rates, and integrated as they are completed... Iterative development is a rework scheduling strategy in which time is set aside to revise and improve parts of the system... The two terms were merged in practical use in the mid-1990s...”

**1.2.2 ¶4** **1**Iterative vs. incremental is a process decision as well. **2**While good, design-focused things can happen as a result of the process, its not really inherent in the design itself. **3**So, it’s an internal, business-function item. **4**Hence, process. **5**This can cause issues, in that process people might be many levels above you, or work for the client, and you can only influence it so much. **6**Having done this myself, and failed to plenty of times as well, I have learned a thing or two about achieving this influence over process:

- 1.** **1**There are good processes and bad one. **2**As an example, incremental design is usually bad. Iterative design is usually good. **3**Do the good one, and avoid the bad one.
- 2.** **1**Get the terms right. **2**Know what you mean, know that everyone understands you, make sure they can tell everyone else about it.
- 3.** **1**See what happens. **2**Does the right thing come out the other end? **3**Does it happen the right way?
- 4.** **1**Evaluate and find root causes. **2**Maybe a bad result is a result of not following the process. **3**Maybe it’s a bad process, though. **4**Perhaps you forgot some critical step that you assumed.

1.2.2 ¶4 continued  
 ...The authors of the Unified Process (UP) and the Rational Unified Process (RUP) selected the term “iterative development”, and “iterations” to generally mean any combination of incremental and iterative development. Most people saying “iterative” development mean that they do both incremental and iterative development. Some project teams get into trouble by doing only one and not the other without realizing it.” That last part is key. In my experience, everyone falls into this last trap, so I ignore the merging of the terms, and think clarifying them to your team is good.

## 1.2.3

### 1.2.3 ¶1

## Procedure – Working With Others

1The details of a process — of naming, forms, locations, etc. — are procedures. 2This is the part where lots of process falls down. 3Procedure is boring, and no one follows it. 4This has previously led me to believe that all procedure is bad, and instead everyone should be free to do what they want.

### 1.2.3 ¶2

1But collaboration does require everyone is on the same page. 2In that case, the quickest way to this is to make sure everyone does stuff in the same way. 3The difference between process and procedure is that “doing stuff the same way” part. 4Unlike assigning resources, where any number of designers can probably do the same job, your procedure will either work or not. 5Failure to follow procedure will result in nothing done, the same thing done badly, or ad-hoc results.

### 1.2.3 ¶3

1Procedure is needed for any workgroup large enough to warrant a process at the business level. 2Of course, a process needs procedures, but even the design part needs them. 3As said above (so don’t restate it) make sure everyone is on the same page re: artifact creation and storage and sharing and that sort of stuff... this means software, and file naming and other boring stuff. 4How to identify these? 5Generally, if its overhead (time not drawing or thinking or talking to clients...) its bad procedure, or needs a procedure.

### 1.2.3 ¶4

1Procedures exist at each level. 2The process guys can worry about their procedures (which forms to submit, and so on), but you need to be concerned with design procedures. 3I’ve worked in a number of large design groups that really needed procedures applied. 4As the organized guy, when I am on small groups that start to grow, I tend to be core to actually creating procedures. 5Here’s the high level categorization of design procedures I have noticed or specified in the past:

- **1Artifact creation**
  - 1.1Everything should be in a common location where everyone on the design team can get to them – today this is probably a

shared drive

- 1.2 Everything should be in a format that everyone on the design team can open and edit – this means the same software tool, or a set of related ones
- 1.3 Artifacts should be completed in a common manner – Document formats, scale and detail level for drawings, style of writing and everything; a style guide and templates are in order here
- **2 Design team**
  - 2.1 Workplace – Is everyone in one location? How to communicate when working from home, at other locations or at the client site?
  - 2.2 Workspace – Is there a team room? Can we design at each other's desks? Is it possible to pod up and make a temporary workgroup?
  - 2.3 Engagement – How do you get the team together? How do you engage additional resources if we forgot something?
- **3 Consumer activities**
  - 3.1 The document consumer or consumers must be identified – Not just people, but the job of each person it will be handed to; several documents might be needed
  - 3.2 Can you talk to the consumers before the design is complete
  - 3.3 How are the final artifacts delivered – Is it emailed out? Is it printed and walked over? Is there a walkthrough of the design with the whole development team?

1.2.3 ¶5 1 Other departments will have their own procedures, by the way. 2 The marketing guys, the sales people, and everyone has procedures for their bit of the processes as well. 3 They will vary in many ways from their process, though they probably don't recognize this distinction.

## 1.2.4 Methodology – How You Design

1.2.4 ¶1 1 The manner in which you work is a method. 2 When this is repeated and codified and applied uniformly it is a methodology.

1.2.4 ¶2 1 Since we're talking about design, methodology

is about activities and artifacts directly related to the design work. **2**Good methodology is about designing well.

**1.2.4 ¶3** **1**These activities and artifacts depend on what sort of design is going on. **2**Software design is totally different from database design, which is different from interaction design. **3**The rest of this document is really about the methodology of interaction design so I am not going to go into any more detail about it here.

## **1.2.5 Analogy – Interactive Design is Exactly Like Making a Pizza**

**1.2.5 ¶1** **1**No, its not. **2**At all. **3**Its also not at all like building a house. **4**Or changing the tires on a car while its driving down the road at 60 mph. **5**Or any other favorite analogy your leadership has used to describe the situation.

**1.2.5 ¶2** **1**Sometimes, this analogy work goes quite a bit too far, and is a fairly bad thing. **2**Processes, for example, are picked up from other fields and misapplied to interactive design. **3**Inappropriate analogy or blindly inherited process encourages inappropriate process and methodology. **4**Houses need blueprints, then foundations. **5**Do websites?

**1.2.5 ¶3** **1**Its time that interaction deign (or even software development) comes into its own, and can be understood without arbitrary analogies.

**1.2.5 ¶4** **1**Another key point in the misuse of analogy is the assumption that everyone else knows what they are doing.

**1.2.5 ¶5** *...I live every day with the vague nightmare that at some point, someone more knowledgeable than myself is going to sit up and pen a massive screed indicating exactly where my work is shallow and fraudulent and rooted in lame, half-assed assumptions.*

— David Simon

**1.2.5 ¶6** **1**Yeah, pretty much everyone has those fears. **2**Individually, organizationally and as whole fields. **3**Architects, and builders and factories and pretty much any industry is chock full of seeking out improvements in their own process due to inefficiencies, inaccuracies and half-assed

**1.2.5 ¶3**  
Standards: just like the print industry has developed over time.  
– Christopher Nemeth  
Designer

**1.2.5 ¶5**  
The quote from David Simon is from Nick Hornby's interview for the August 2007 issue of *The Believer*. You can read it here: [www.believmag.com/issues/200708/?read=interview\\_simon](http://www.believmag.com/issues/200708/?read=interview_simon). The Wire was among the best shows ever on television, so I believe everything Simon says. And he says it all so well.

implementation. 4You can be an expert in your field and believe it is able to stand on its own.

1.2.5 ¶7 1That said, there will be one or two small analogies in this text. 2Just don't over use them.

1.3

## Systems of Information

1.3 ¶1

1Though design certainly takes place on all sorts of products, the subject here is interactive systems. 2Mostly, we mean computer-based systems, such as websites and programs with graphical user interfaces. 3They can certainly be applied to any number of other systems of information like interactive voice response systems, handset menus or printed materials.

1.3 ¶2

1Fundamentally, presentations of any sort can be distilled into being information presentation systems. 2I'll go into more details on this later, but for now, just believe me.

1.3.1 ¶1

The book *Information Design*, published in 2000 by the MIT press, is an anthology of writings on the topic, strongly reflecting UI design and the interactive media just emerging at the time. But Robert Jacobsen does such a good job of selecting and editing it together the collection reads like a single narrative thread. Among my favorite parts is that the cover is not an illustration, but an IA diagram of the contents of the book. It believes in its own topic that much. Brenda Dervin's article is just one of many, and probably the most academic; I also want to acknowledge the decades of work she put into this, which I have simply absorbed and put down again as a few sentences. There is no way I could come up with these fundamental truths myself.

1.3.1

### Making Sense of Information:

1.3.1 ¶1

1Brenda Dervin (in her article for the book "Information Design") argues that information design has always been with us. 2Assuming information is a thing that can be manipulated, the (western at least) view of information to those who wrote about it, in chronological order has been that information:

1. Describes an ordered reality.
2. Describes an ordered reality by can be "found" only by those with the proper observing skills and technologies.
3. Describes an ordered reality that varies across time and space.
4. Describes an ordered reality that varies from culture to culture.
5. Describes an ordered reality that varies from person to person.
6. Is an instrument of power imposed in discourse on those without power.
7. Imposes order on a chaotic reality.

1.3.1 ¶2

1And now, the post-modern sense is that: Information is a tool designed by human beings to make sense of a reality assumed to be both chaotic and orderly.

1.3.1 ¶3

1This considers information design to be "sense-making," or making sense of information thru communication of information, or designing how that information gets to the user. 2This relies very much on design as making sense of infor-

mation requires that you not be overloaded — so not all information is presented at once — and that the information be processed, or others input be valuable in achieving that understanding.

### 1.3.2 Information Design?

#### 1.3.2 ¶1

If you aren't following my concept of information and information design, try the movie on this page: [infosthetics.com/archives/2009/02/what\\_is\\_information\\_maya\\_movie.html](http://infosthetics.com/archives/2009/02/what_is_information_maya_movie.html) It does a brilliant job explaining how information is not the medium but the message.

1.3.2 ¶1 1Though terminology in these fields is sometimes troubling, I call the active design of this stuff “information design.” 2Other people call it something else, or define information design in slightly different way, but don't worry about that and take my word for it for the rest of this discussion. 3It's easy to remember, as its about “designing information.”

1.3.2 ¶2 1As we just said, information design informs people so they can make their own information senses. 2By information senses I mean a way of perceiving the information, or their own mental model. 3Information — created information — is purely for this communication purpose. Remember above, it's not the reality itself. 4It is meta-information, or synthetic information to describe reality, not the ‘real’ data that nature uses to describe itself. 5This means that information design does not address adding order to reality, and is about presenting information to the end user so they can understand and act upon it.

1.3.2 ¶3 1As the information is meta-information, information design is metadesign, design of — or about — design.

1.3.2 ¶4 1Another way to express this is to jump back to the Good Design discussion at the beginning. Form is tightly coupled to function. 2Poor formatting, inappropriate design, or excess decoration can impede the understanding of information. 3The corollary is that good or appropriate formatting can assist understanding of information. 4In short, form can lead to meaning.

1.3.2 ¶5 1This should be pretty simple to understand as it relates to the presentation of charts of data. 2But it works for all information senses. 3As an extreme, take an electronic control system for an aircraft. 4Inputs by the user change the orientation of the aircraft. 5The user can detect these changes by looking outside, but you cannot

1.3.2 ¶4 s4  
Probably need to reference Tufte here.  
– Christopher Nemeth  
Designer

Yes. Something so ingrained I forgot to say it. Next time.  
– SH

1.3.3 ¶1  
Chiara Fox is a compelling speaker, who clearly states big ideas better than most others. Most of her ideas, and all of her talks, are listed on her site at: [chiarafox.com](http://chiarafox.com)

1.3.3 ¶2  
The definition for taxonomy that I used is from the introduction of the Wikipedia article. It complies with all the others I've seen, but is a bit better written than most, so I used it instead.

1.3.2 ¶6  
1As I said, this is an extreme example, but its still information design. 2Using information (the control stick feedback) to make sense of a system (the orientation of the aircraft).

### 1.3.3 Truth or Information?

1.3.3 ¶1  
1Chiara Fox was up on stage at an Adaptive Path seminar in 2007 and went briefly into an aside on what taxonomy really means. 2I had never looked it up before.

1.3.3 ¶2  
1Taxonomy is the practice and science of classification. 2The word comes from the Greek *τάξις*, *taxis*, 'order' + *νόμος*, *nomos*, 'law' or 'science'.

1.3.3 ¶3  
1Which is interesting. 2I had been using it in the more common way to mean any remotely hierarchical classification method. 3Apparently, this bugs the hell out of library sciences people, and the type of folks who discover new insects. 4At first I thought they were just being nit-picky, but then I started thinking that this distinction is useful.

1.3.3 ¶4  
1Taxonomies are only of actual, naturally-occurring items (originally just living things). 2So, although there are ways of organizing actual items, the item itself is not modified in any way.

1.3.3 ¶5  
1Information design takes all the rest of the information in the world — words, recordings, photos, data streams, relational databases — and uses a methodology and series of best practices to make the information useful and understandable. 2Information design can be applied broadly, but it cannot be applied universally. 3You cannot really design how a tiger is expressed to a viewer (or victim).

1.3.3 ¶6  
1However now consider a taxonomy, in the strict sense, where it is a hierarchy of animals to each other and their higher classifications. 2Naturally,

that can be designed, or presented in numerous ways, depending on the intended need. **3**How can this be? Because the taxonomy itself is something created by people. **4**Even when expressing some fundamental truth, it's not the true nature itself, but an interpretive understanding.

**1.3.3 ¶7** **1**Why does this matter? **2**It might seem to be purely navel-gazing philosophical discussion; who cares about the information, if all we're doing it organizing it. **3**Well, one important difference. **4**If it's real information, then the data (and its format) is inviolable. **5**If its been fabricated or modified, then its up for grabs.

**1.3.3 ¶8** **1**What do I mean? **2**Well, go over to your bill pile. **3**How many of your bills refer to you entirely in all caps. **4**My name is Steven Hooper. **5**I understand in some foreign countries they like to address with the last name in all caps, but generally my name is not STEVEN HOOBER. **6**Not to mention that the rest of my address, the info about my bill type, etc. is all caps. **7**And don't get me started on date formats. **8**Why is this? **9**Because at those companies the database is old and/or badly designed, and there is a general acceptance that the data store is a magical land.

**1.3.3 ¶9** **1**Wrong. **2**If the information is strange, hard to read or useless no matter how pretty the surrounding box, it has still failed. **3**Modifying the data for presentation, or better yet making sure the right data is stored in the right way, is well in the bounds of what I consider to be information design, and an information designer's tasks.

**1.3.4 ¶1**

Tiziana Haug once said design is never right or wrong, just more or less appropriate. I have no idea where she said this, but it was mentioned here: [www.transism.com/weblog/-design/2005/12/](http://www.transism.com/weblog/-design/2005/12/).

## **1.3.4 Contextual Appropriateness**

**1.3.4 ¶1** **1**Designer Tiziana Haug once said that design is never right or wrong, just more or less appropriate. **2**Think about how often a great and beautiful design simply does not fit some specific situation. **3**That is what I mean by "inappropriate design."

**1.3.4 ¶2** **1**Context is what we are talking about here. **2**How to perform this, and guidelines on what to execute for different situations, is discussed in some detail later, but the concept is important

here.

- 1.3.4 ¶3 1Typically, there is a core set of information to be presented. 2The mere fact that we have jobs as designers implies it can be, or even should be, designed. 3That means there are multiple methods of presenting the information. 4How do we come to the right presentation? 5By considering the audience, their needs, the use to which they will put the information and any resulting ideas they come up with, the display context, the environment, the time of day, their interaction with others, their individual abilities or needs, and many other factors.
- 1.3.4 ¶4 1What do we really mean, though? 2For interactive systems, the most obvious examples are things like device appropriateness. 3Mobile devices, like phones, can view web pages. 4Often, however, they really should not. 5Even transcoding (to make the page ‘fit’ on a mobile screen) is a poor solution. 6The best application providers offer custom versions of their site or tool for different display environments.
- 1.3.4 ¶5 1Another example, less obvious to most users, is a billing system. 2Your printed bill and the online version of your account are all generated from the same basic set of data. 3But this is massively simplified. 4Customer care reps get another view of the exact same system — the same core information — with much more information. 5Some of this is notes or ratings about the customer you would never want them to see directly.
- 1.3.4 ¶6 1Even then, the care rep probably cannot see the database directly. 2It’s still too complex, its organized in a way to make the software work well, and the presentation is (hopefully) designed for the way people use the information, and there is too much chance to break things. In each and every case of design, a decision was made as to what information should be presented, how much it should be presented, how it should be presented, and most of all, what should be hidden.

### 1.3.5 Design Responsibility

1.3.5 ¶2-4

The Best Buy court case quote is from eweek: [www.eweek.com/article/2/0,1759,2136241,00.asp](http://www.eweek.com/article/2/0,1759,2136241,00.asp) "Connecticut Sues Best Buy for Deceiving Customers" Evan Schuman 2007-05-24, and seen in many other sources.

- 1.3.5 ¶1 1Designers have fearsome power. 2The truth can be obscured by inappropriate design, even if you aren't trying to lie, cheat or steal from your users. 3This is a case case that brought Best Buy to court in 2007:
- 1.3.5 ¶2 1*After months of investigation, the state of Connecticut sued Best Buy on May 24 and accused the chain of tricking customers with two identical-looking Web sites in its stores, with the only difference being that one had higher prices...*
- 1.3.5 ¶3 ...1Initially, Best Buy said the identical design was used for both sites to save on design and programming costs. 2On May 24, Best Buy added that it designed to comfort — not confuse — their customers.
- 1.3.5 ¶4 1*"We used the same Web site platform for these in-store kiosks as we did for our national Web site to ensure that customers familiar with the national Web site could easily navigate the in-store kiosk," Busch said, in a prepared statement.*
- 1.3.5 ¶5 1Let's assume that no one did this maliciously (and if it does, that opens a whole can of ethics we don't need to get into). 2And if you have ever worked for or with any large company (especially a technology-heavy one) that at each step there were probably good reasons for this. 3Time and money can be saved by sharing design and platforms. 4Marketing has approved a design, and likes it. 5Branding means there is only so much leeway in design anyway. 6The change may have been last minute.
- 1.3.5 ¶6 1But fundamentally, design is what failed the end users and any designer working on this should have been able to perceive the effects and try to do something about it (maybe someone did, and got shot down; I wasn't there).
- 1.3.5 ¶7 1Do you want to lie to customers that much? 2What lawsuit-level risks are you taking with your customer's information or the presentation of that information?

## 1.4 Information Design as a Guiding Principle

- 1.4 ¶1 1A guiding principle. 2Big concept. 3And to a certain degree I do mean it as a way of living your life. 4Accept it and you will become a more natural designer.
- 1.4 ¶2 1But here I am really talking about creating a set of guiding principles to lead you through any specific product, project or design.

### 1.4.1 Systems of Designers

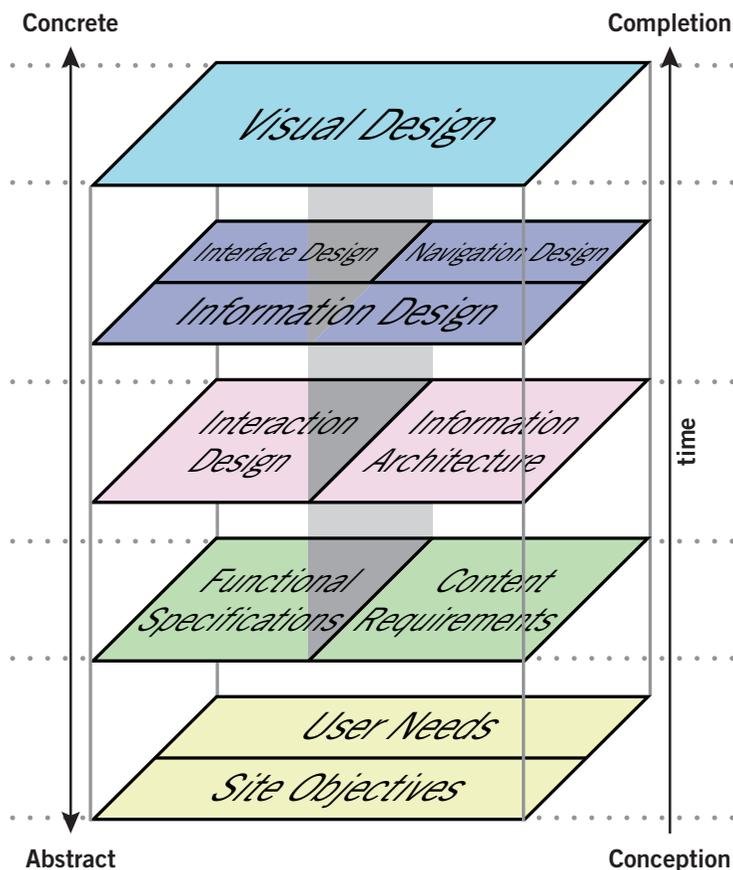
- 1.4.1 ¶1 1Information design, within the confines of interactive design, is the practice of designing a usable, understandable presentation method for hypermedia and other interactive products. 2Information designers accomplish this by careful attention to pre-determined graphical standards, system interaction needs and product requirements. 3Attention must also be paid to simplicity of construction and development, integration, testing, maintenance and upgrades.
- 1.4.1 ¶2 1Information Design should never stand alone. 2It's part of a larger, multi-disciplinary user-experience and graphic design team. 3Design of products will require the integration of business requirements with user studies, graphic artists, coders and information design methods. It also will need to be implemented by some other team, who should be engaged to make sure it can be built.
- 1.4.1 ¶3 1The information design process, standard notations and documentation will be used to assure a consistent and thorough communication of the design to product owners, and downstream implementation teams.
- 1.4.1 ¶4 1Our economy depends on growth through increased efficiencies (and in most places, increased sales). 2The current systems — especially interactive ones — are fraught with gross inefficiencies. 3Well-designed systems inspire correct use, are faster and generate fewer errors. 4Usability and interaction design focus on creating well-designed systems that foster quality, acceptance, efficiency and productivity.

## 1.4.2 Interaction Design, Information Architecture, and Everything Else

1.4.2 ¶1 1You are going to note that I talk a lot about Information Design, and not so much other terms that are common in the industry.

1.4.2 ¶2 1Some of my friends tell me I am generally contrarian, and disagree with everyone else as a matter of course. 2I think that goes too far, but I am going to go ahead and disagree with the constantly reproduced, practically industry-standard layer diagram Jesse James Garrett published in *The Elements of User Experience*.

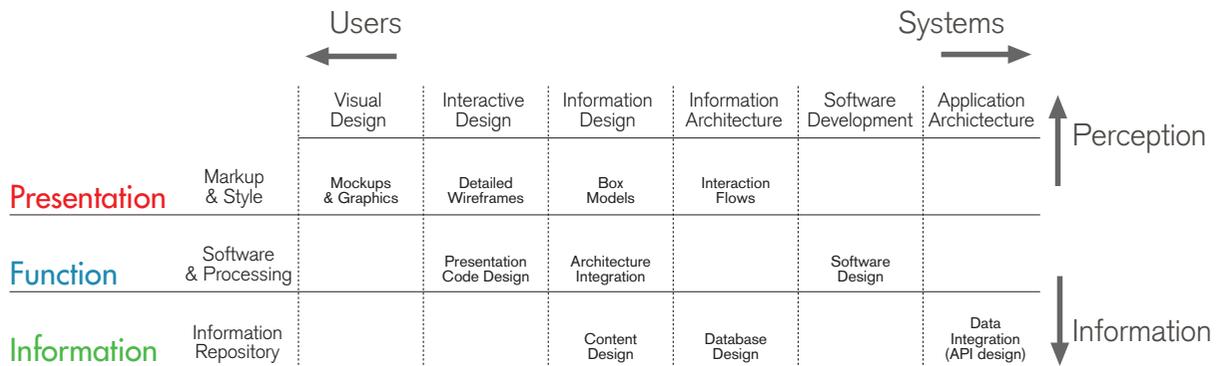
1.4.2 ¶3 Jesse James Garrett's diagram (reproduced by permission) is cropped from his original diagram available at [www.jjg.net/elements/](http://www.jjg.net/elements/), and as printed in his book *The Elements of User Experience: User-Centered Design for the Web*, New Riders Publishing, 2002. This is an excellent book, and if you can't be bothered, at least pore over the free PDF for a while. It's still a plenty good model for designing and understanding design.



1.4.2 ¶3 1This is actually just a portion of it. 2You can get the full diagram from his website at [www.jjg.net/elements/](http://www.jjg.net/elements/).

1.4.2 ¶4 1Instead, I take that philosophical view of information from the previous chapter to its extreme-

ly extended conclusion. **2**Everything is about information processing, information organization and information presentation. **2**So, I have flipped the chart and added something specific to the other axis.



1.4.2  
Another example of such design frameworks is the recent discussion of them in the middle of this [www.christianlindholm.com/christianlindholm/2009/02/mobile-experiences-the-new-paradigm.html](http://www.christianlindholm.com/christianlindholm/2009/02/mobile-experiences-the-new-paradigm.html) This is troublesome. Bling implies ornament, when that level should be about making the content and function clear. And his bottom level seems to be “everything I can’t influence.” Sadly, the fact that he has one at all puts him ahead of a lot of designers, who have put no conscious thought into their conception of design at all.

- 1.4.2 ¶5 **1**You might note that I have broken it up into these rather IT-centric slices. **2**This is not from some secret love of IT process, but because it’s the right way to address the three layers of information, function and touch for interactive systems — it’s not too different from ways of looking at physical objects, actually. **3**(And anyway, IT never seems to implement stuff correctly, and is always throwing software into the presentation or data sides, for example.)
- 1.4.2 ¶6 **1**This is also important not just from an implementation or structure point of view, but from a design point of view. **2**Without good data, there is no point in presenting it. **3**Without function, you cannot make the information easy to understand.
- 1.4.2 ¶7 **1**And that leads lastly to the arrows on the sides, which I hope explain the whole concept. **2**Along the top, while all design impacts systems and users, that on the left is much more user-impacting, while that on the right is much more system-impacting. **3**Interaction is the relationship between the user and the system, so all those UX-centric portions are in between the user and the system itself.
- 1.4.2 ¶8 **1**Note that other practitioners will do, or help, with many of these other steps. **2**Usability researchers, information architects, visual designers and others all have their specialty areas,

and if you can engage them, do so. **3**Ideally, get everyone on the same process page so you are all speaking the same language before proceeding with a project.

- 1.4.2 ¶9** **1**There is one thing that Garrett's chart does which I can't quite fit in there myself. **2**No, not the pretty colors and isometric view. **3**He addresses the conceptual stages on that bottom row. **4**While the process I follow does have specific procedures and artifacts at these stages, I am not sure the organization of an interactive system can be meshed into the process cleanly enough to depict the two at once. **5**So far, I am leaving it out, though I might add in something later.

**1.4.2 ¶9 s3**  
Is it an overarching principle that "conceptualization" is not a stage but the very evolution of the idea throughout the entirety of the process?  
– Christopher Nemeth  
Designer

### **1.4.3 Hypermedia or Application?**

- 1.4.3 ¶1** **1**One thing I really did like about the chart reproduced above by Garrett is that he recognized the distinction between software-like web interfaces (on the left) and brochure-like, hypertext web interfaces (on the right).
- 1.4.3 ¶2** **1**I totally believed this for several years. **2**Partly because I love classifying things. **3**Anything that can be categorized is good. **4**Early versions of my process made a solid distinction between the two styles, and there were different processes for each of them. **5**Almost immediately, however, I started running into products that had significant elements of both. **6**The more I worked on it, the more I would come up with designs that blurred the line between the two. **7**Ajax, interesting uses of Flash and other technologies are increasing the ability of web interfaces, especially, to work in new and dynamic ways.
- 1.4.3 ¶3** **1**So, I have worked on all these processes and artifacts to work for any sort of interactive element. **2**Though I have not done any of note, I cannot think of a reason it wouldn't (at the most abstract level) even work for most or all physical device interfaces.

## 1.4.4 Gathering, Collecting and Starting to Analyze

- 1.4.4 ¶1 1Whether you are redesigning a new system, or starting with a clean sheet of paper, before getting to any drawing or technical details, you need to gather the most basic information about the product, concept and users. When you get to design, this will be critical to make sure you are building for the correct context and scope.
- 1.4.4 ¶2 **Collect**
- 1What is the product?
  - 2What is its one main purpose?
  - 3What one problem or concern does it solve?
  - 4Who will use this product?
- 1.4.4 ¶3 **Systems, Processes & Integration**
- 1Is this product similar to any existing product or feature? Can this product be used instead or leveraged to ease implementation?
  - 2How is this product intended to be integrated with the existing larger system?
  - 3How is this product intended to be integrated with global or physical systems?
  - 4How is this product intended to access new or existing data?
  - 5What would be the impact of changing this expected architectural integration?
- 1.4.4 ¶4 **Benchmark**
- 1What can be learned from other (outside) systems that have implemented or attempted to implement similar features?
- 1.4.4 ¶5 1Do pay attention to business constraints, because you need to build a product that the business needs, as well as the users. 2Pay attention to technical constraints that are at a business level. 3Like, if the company has made a decision on a platform, that's important, and avoiding it will not get you to a workable solution. 4Don't worry about stretching the technology in the details, though. 5For example, if your developers haven't done Ajax before, don't worry about that. 6Good ones will want to learn, or maybe some vendor can help.
- 1.4.4 ¶6 1First, you need information. 2We're not going

to talk in detail about how to run user studies, or persuade the business to give you this stuff, but its important. 3You need this sort of information to really do the design stuff correctly. 4Be aware of the difference between types of research. 5Marketing will often bring fairly un-robust research, or internal opinion as data to drive requirements. 6Be sure you drill into this so you know where its coming from.

1.4.4 ¶7 1It is also important to understand this information gathering is before you can do most user studies. 2First, you are trying to understand the client's space and intent.

1.4.4 ¶8 1The methodology as guidance semantics is important here. 2These are just guidelines. 3Your product or industry may not need some of these, or might very well need more information. Think about these factors before you get to gathering it, and as you find information

1.4.4 ¶9 1Now, we'll go over these items in detail.

#### 1.4.5 **Collect**

1.4.5 ¶1 1Its always best to get this information directly from the business owners. 2Who this is will vary a lot (and could be marketing, product development or even business analysts), and will be up to your knowledge of the organization. However, keep the group small. 3More than about 5-6 people in the room, total, and you will start getting more caught up in conversation than information gathering. 4If more than that many are absolutely needed, it's a good hint that the product is too complex and needs to be broken up or narrowed even as a basic concept.

1.4.5 ¶2 1And that has brought up a concern with a number of folks I work with. 2You only want to interview people in a group very carefully, with a very good moderator, to avoid contaminating or suppressing any one user's opinion. 3But for product owners, the product vision is often spread among a group, and no one person has it all in their head; group contribution will be needed. 4Be careful of politics, and try to have reasonably senior people, but all at broadly the same level. 5You can encounter yes-man issues if

the big boss of most or all of the others is in the room.

- 1.4.5 ¶3 1This should probably be an immersive session, a few hours (two 4 hour sessions, over two days, works well) to get all the information down. 2Use all your info gathering skills and bring lots of post-its, paper and markers. 3Computers and, if possible, phones are off. 4Close the door, and get everyone to focus.
- 1.4.5 ¶4 **What is the product?** 1This can also be stated even more briefly as “what is it.” 2This should be one sentence, with as few commas as possible, when you are done. 3Everyone in the room will have at least five ideas, so gather them up. 4The best way is to have everyone write them down, then the moderator (you) will group them and see if everyone can agree on what group looks most important. 5One usually naturally falls out.
- 1.4.5 ¶5 1This is not the problem statement, or a solution in that its addressing a problem. 2It should be very, very simple. 3Like, a web-based portal for viewing billing information.
- 1.4.5 ¶6 **What is its one, main purpose?** 1This is still not a problem, or solution statement. 2It speaks to function, or tasks the user will perform, without being an actual task map. 3One item only, so the same exercise as above will help you get there. 4Think of tasks like “find and book airline flights” or “change your user ID.” 5It may or may not be worth stating the domain, depending on the company or group.
- 1.4.5 ¶7 1Larger products might seem to need more than one purpose, but its still important to give everyone a single focus point. 2Large products like entire account management systems can be challenging. 3Try to find something unique, to your customers, the company or the way this individual product will be built. 4Even the act of discovering this one purpose can be very illuminating to the product owners.
- 1.4.5 ¶8 **What one problem or concern does it solve?** 1This is, finally, the problem statement. 2If there is no problem explicitly to be solved, and its more of a business opportunity, express those as

problems, instead.

- 1.4.5 ¶9 1Again, I have to emphasize that its one problem only. 2If the business brings problems in, they will bring lots of them from various research, anecdotes, media coverage, etc. 3This is a time when careful use of your discovery methodologies will help to find relationships and core problems.
- 1.4.5 ¶10 1If there is no actual problem, more analysis will be needed, and some explanation to use client as to what we mean by problem, or customer need.
- 1.4.5 ¶11 **Who will use this product?** I think of this as simply “who are the users.” 2Your clients might be best served by being asked “who suffers from this problem?”
- 1.4.5 ¶12 1If you ask who the product is for, the default answer is almost always either “everyone” or “every customer” and then you have to drag details out of them. 2Asking it in the sense of the problem statement often helps narrow it down. 3It should be possible to make a single statement about customers having a problem or need.
- 1.4.5 ¶13 1To guide you better through actually getting these answered, I am including a sample document I have used to get these answers. 2The phrasing describing what we want is typical of what I say in person, but note that this is something intended to be emailed our (or distributed over Survey Monkey) for those times you simply cannot get all your product owners in a room.

This information is to be filled out by yourself, individually. Not only should you not collaborate with others, but do not worry about what others may answer. Your concerns, thoughts and phrasing will be different from anyone else, so there is no such thing as an answer that duplicates what someone else says.

#### **What is the product?**

You are in an elevator with a chief executive of your company. He asks, “what are you working on these days?”

In one short sentence, using plain (non-technical) language, explain **what the product is**.

Now, do it again. Try to answer at least three times (total), in different ways. Make sure each answer is true, even though no one answer can completely cover all the features and capabilities.

**What is its one, main purpose?**

Eventually, a complete list of features will be created, and from this the design of the screens, the software and the content management systems will flow. Before this, however, there are high-level categories of features (this is similar to business requirements, business needs, etc.). This will be considered by the end user to be the purpose of the product. These are the features you have in mind when discussing the product.

**Pick a single feature of the product you think is critical and express it in as few words as possible.** 1-2 word phrases are perfectly fine (“receive cards”); these do not need to be complete sentences. Do not consider technology, UI, wording or other content at all.

Now, answer it again. You may answer as many as five times in total. Do not restate any points; each one should be unique.

If the domain (i.e. the mobile phone application) is obvious or always the same, there is no need to state it each time.

**What one problem or concern does it solve?**

Products are pursued as a result of a business opportunity, or a business problem. Consider any opportunity to be a “problem” in the sense that its something the company is not pursuing (so a missed opportunity for now).

Re-read your definitions above, and answer why you want to build such a product:

**What is the one problem the product will solve?** Do not state how it will be solved, just what the problem itself is.

Please limit your statement to a single, short sentence. Avoid listing items, asides, definitions or prevaricating phrases (e.g. “or other things”).

**Who will use this product?**

Instead of trying to design the product for everyone, we will be focusing on feature sets, and interface designs that meet the primary

needs of a small but focused set of users. These should not be market segments as they exist today; instead consider the manner in which you expect these customers to use the product. Think of:

- Who they are
- Where and when they use the product
- What sort of items they search for and send (or, receive)
- Who they are sending the items to (or receiving from)

**Who do you see as the core users of this product set?** You may answer either as “the most common” or your “most desired” users.

If you have several users in mind, feel free to include them all, but make each one an independent statement.



## 1.4.6 Systems, Processes & Integration

1.4.6 ¶1 1A key manner in which UX can help a business is in considering design holistically. 2Do not assume, or allow the client to assume, that the project is free-standing. 3It will reflect and interact with the company, other products that already exist, internal systems, competing and collaborating companies, the marketplace and the user’s environment in general.

1.4.6 ¶2 1This is research that might have already been done, or which might be able to be done by others, but its something you will need. 2There is no specific need to do this collaboratively with the product owners, though you may be able to sometimes. 3It’s just an analytical technique for the designer to have on hand.

1.4.6 ¶3 **1Is this product similar to any existing product or feature? Can this product be used instead or leveraged to ease implementation?**

- 2The Product Manager will know some of these, especially if its an upgrade from another product. Get any documents about that implementation, and go try to use it.

1.4.6 ¶4 **1How is this product intended to be integrated with the existing architecture?**

- 2All products operating in a larger system need to be integrated to some degree.
- 3As many features as possible should be shared for modified from existing features of other products or the global system.
- 4The intended integration method may not be the best. Evaluate it and use your experience and the process to determine the best integration method.
- 5If there is no integration plan from the business owners, create one with them.

#### 1.4.6 ¶5 **1How is this product intended to be integrated with global or physical systems?**

- 2Determine the method by which this product will be delivered, ie: wired web, wireless disk.
- 3Apply any available heuristics to the expected delivery method.
- 4Take into account the physical environment the user will be in when interacting with the product. Eg: wireline web systems can be accessed from home, work, public locations like libraries, friend's houses and so on.

#### 1.4.6 ¶6 **1How is this product intended to access new or existing data?**

- 2Dynamic, data-driven systems need to have their data access schemes developed to a certain degree up front. Discuss with architecture and other groups as needed.
- 3Try not to repeat creation, storage and retrieval of data already stored in other systems.
- 4Get an idea of which fields will be needed or already exist, and their limitations so the presentation layer accurately reflects the information and does not need to be modified to fit later.

#### 1.4.6 ¶7 **1What would be the impact of changing this expected architectural integration?**

- 2If you feel a change is required relative to the requirements documents, find out if this change will have any impact to the implementation of the product. Small changes to the presentation layer is no big deal, but if you recognize data schema changes, this

will impact development.

## 1.4.7 Benchmark

### 1.4.7 ¶1 **1What can be learned from other (outside) systems that have implemented or attempted to implement similar features?**

- 2Some information may be provided by the business owners as far as existing services they wish to emulate or compete with.
- 3These services should be evaluated as much as practical to see what does and does not work. The UR department could assist with this.
- 4Do not disregard similar systems that failed or needed serious changes after implementation. Learn from others' failures, as well as their successes.

1.4.8 ¶3, 4  
Morphology definitions are from *thefreedictionary.com*, though this is unlikely to be the organization that developed it

## 1.4.8 1.4.8 Understanding Design Through its Morphology

1.4.8 ¶1 1Once you have this core information you can start analyzing it. 2This is important. 3You cannot do this next phase out of the blue. 4You need to have a grounding in what the product is going to be, what it's supposed to do for the business, and what it is supposed to do for the customer.

1.4.8 ¶2 1Now, we're back to semantics. 2I'm betting you don't know what morphology is. 3I didn't until I went looking for the right word. 4Instead of making you look it up, I'll just tell you.

1.4.8 ¶3 **1Biology:** *The branch of biology that deals with the form and structure of organisms without consideration of function.*

1.4.8 ¶4 **1Linguistics:** *The study of the structure and form of words in language or a language, including inflection, derivation, and the formation of compounds.*

1.4.8 ¶5 1So, a merged, generic definition might be:  
2A manner of studying and defining structure and form, including the manner in which they were derived, but without consideration of function.

1.4.8 ¶6 1So, how does this apply to interactive systems?  
2Well, it's obvious to me. 3But I am the one writing, so I'll explain it. 4Information design, you will recall, is about form leading to mean-

ing. 5Interactive elements might seem to be functional, but in another mode of analysis, they are simply extensions of methods of presenting information, or the form and structure of the information presentation.

- 1.4.8 ¶7 1So, if we disregard the true function (the software behind it all) then it should be reasonably easy to discuss these. 2A taxonomy (or rather, a classification) of structure might help also, though. 3This one has worked for me:

Coherent	Discoverable
Inclusive	Engaging
Malleable	Responsive
Secure	Protective

- 1.4.8 ¶8 1Though I did have to add the last two after a while. 2I might still have missed something (though I have used this for several years now) so if you need more, or want to change some, do so.

- 1.4.8 ¶9 1I like the theory of letting designers make their own definitions for these, but that takes a long time, so instead I'll just tell you what I was thinking when using these labels.

- 1.4.8 ¶10 1**Coherent** – The product and information should easily comprehensible and of a single mind. 2Different parts should all work in tandem to support each other. 3Each part should be immediately understandable to the user base without undue assistance, help, explanation or training.

- 1.4.8 ¶11 1**Discoverable** – This is about wayfinding, but expanded to general understanding of information senses, not just navigation. 2The product, and all its functions, should be able to be found, or discovered, without training or big yellow starbursts pointing at them.

- 1.4.8 ¶12 1**Inclusive** – The user should feel they are a part of the process, in the sense they are part of a community. 2Online enabled social activities are perfect, but remember a sense of belonging can be achieved with simple steps like presenting relevant information, and greeting them accurately.

- 1.4.8 ¶13 1**Engaging** – I think this covers it: “The two

1.4.8 ¶13

The quote I originally found was misattributed to William Makepeace Thackeray, so it took me a while to find the reference. The full quote is “In this work are exhibited in a very high degree the two most engaging powers of an author. New things are made familiar, and familiar things are made new.” - Samuel Johnson, “The Life of Alexander Pope” from *Lives of the English Poets* (1781)

most engaging powers of an author are to make new things familiar and familiar things new.” — Samuel Johnson.

1.4.8 ¶14

**1 Malleable** – The user can change their environment. **2** Customized portals are the ideal example, but smaller cases with just as much user satisfaction abound. **3** Remembering view states and search history are good targets. **4** True customization is very poorly used so keep that in mind.

1.4.8 ¶15

**1 Responsive** – User actions must have immediate and expected consequences. **2** This doesn’t mean the completed action must be instant, but something must happen to tell the user their input is accepted. **3** This is both in the details (click states) and in the overall communications (response emails to your order)

1.4.8 ¶16

**1 Secure** – The system must keep the user’s information safe and accessible only to those that have authority. **2** Identification, authentication and authorization schemes must be present to protect the information. **3** It also must be clear to the user that these steps have been taken.

1.4.8 ¶17

**1 Protective** – The user is prevented from making mistakes, especially those that can violate security. **2** If your site ever presents “you don’t have the right credentials” errors when certain nav items are clicked on, or fires alertboxes requiring phone numbers be in a specific format, its not being protective. **3** Pre-emptive error-avoidance — through design — instead of error notification, is the key here.

1.4.8 ¶18

**1** To analyze a factor of the design, take one from each column, and define what a system with those two attributes might be like. **2** Like this:

<b>Coherent &amp; Discoverable</b>	Users must be made aware of the boundaries, scope and meaning of the system, and their current logical and functional location in the system.
<b>Inclusive &amp; Engaging</b>	Users are invited to feel they are part of a community or activity and that they have created or own a part of the system.
<b>Malleable &amp; Responsive</b>	The system environment is customized on entry and modified as requested based on user needs, tastes, requirements and preferences.

1.4.8 ¶19

**1** And so on. **2** Now you have your functional

definitions. **3**Defining every possible crossed pair will take a long time and lots of space, and you might disagree, so I am not providing those here.

**1.4.8 ¶20** **1**With the pre-work information, you should be able to go through the first chart, and determine which of them are most important. **2**Use any method you want to determine this (do it yourself, engage a team, engage the business owners). **3**Then make matches and list them out. **4**Try to do the top three or so. It's okay if you only have one from one of the columns. Not ideal, but its okay.

**1.4.8 ¶21** **1**Let's look at a sample case:

<b>Coherent &amp; Discoverable</b>	Users must be made aware of the boundaries, scope and meaning of the system, and their current logical and functional location in the system.
------------------------------------	---

**1.4.8 ¶22** **1**Now take the default definition, and modify it to be be accustomed to your product, product, users, etc.

<b>Coherent &amp; Discoverable</b>	Users <i>customers entering Manage</i> must be made aware of the boundaries, scope and meaning <i>limits &amp; capabilities</i> of the <i>registration</i> system, and their current logical and functional location in the system <i>that registration is simple, quick and will improve their ability to access Manage.</i>
------------------------------------	---

**1.4.8 ¶23** **1**Since that was hard to read, here it is with all the strikethroughs removed:

*“Customers entering Manage must be made aware of the limits & capabilities of the registration system, and that registration is simple, quick and will improve on their ability to access Manage.”*

**1.4.8 ¶24** **1**This is going to be a little clunky, and probably be a run-on sentence like I just wrote. **2**You can clean it up if you want, but it's not important. **3**No one has to see it outside the design team, so only clean up as needed so you understand it, or if you do share it with others.

**1.4.8 ¶25** **1**Now, look at all the phrases you made. **2**One should jump out at you as being the best match to what the product is supposed to be. **3**If it doesn't, just deconstruct it.

Customers entering Manage...	Find the best match for the core audience.
------------------------------	--

...must be made aware of the limits & capabilities of the registration system...	Are limits and capabilities the primary focus, or is it something else?
--	---

1.4.8 ¶26 1And so on.

1.4.8 ¶27 1Every time you start this work, and regularly during it, remind yourself of the key attributes gathered under the Collect phase above. 2This is especially important when working as a group. 3If new brainstorming begins to find new key attributes, and there is time to reconsider, stop and go back to that phase, then start the morphology analysis over again.

1.4.8 ¶28 1By proceeding one step at a time, and keeping in mind previous steps, you will be meeting the stated objectives, can trace design decisions to specific goals, and can use this information to analyze and modify the design as needed, after it launches.

## 1.4.9 Creating Design Objectives

1.4.9 ¶1 1This is all going to lead to your first design artifact. 2Yes, you have been writing down and keeping, and maybe even distributing some other work, but they have really all been data collection and analysis. 3This is a design item.

1.4.9 ¶2 1Its not, however, something drawn so it is discussed in detail here, instead of in the artifact creation section later.

1.4.9 ¶3 1First, we just write things down. 2The design objective well help keep you, and the whole team, on the same page throughout the design. 3Pretty much every time you draw a new element later, you should make sure that it abides by the design objectives. 4Personally, I always paste these into the front of the deliverable design document so everyone can read it, then view the designs in the context of these objectives.

- 1.4.9 ¶4 **Structure of a design objective:**  
1Because of a business (or possibly, technical) objective, and/or to solve a known problem the interface must address user needs in a morphologically described sense by:
- 2Doing something specific
  - 3And some other specific items
- 1.4.9 ¶5 **Here's a real example:**  
*Manual registration is a requirement for improved security, as well as within our current technical environment, but is a known barrier to entry for customers. To assure as many customers as possible complete registration or conversion, the process must be as seamless as possible:*
- *As few fields as possible should be presented*
  - *Whenever information is systematically available, it should be pre-populated*
  - *Information should be grouped by task*
  - *Labels and descriptions should be clear, truthful and use standard terminology*
- 1.4.9 ¶6 1Let's deconstruct it, and see how well it does the job, and how I got there.
- 1.4.9 ¶7 **Business objectives:**  
*Manual registration is a requirement for improved security...*
- 1.4.9 ¶8 1This is important, but ought to be easy. During the requirements gathering, you should have determined what these items are. 2Hopefully, there is one real issue you have extracted from this. 3See the information gathering, and especially the Collect phase, above.
- 1.4.9 ¶9 **Technical constraints:**  
*...as well as within our current technical environment,...*
- 1.4.9 ¶10 1These are real. 2Don't go looking for them, but if they exist, don't ignore them either. 3This will be usually big items, like platform constraints (as here). 4Don't harp on them, though, and make it look like you are blaming the technical teams. 5Everyone will know it, and if not, they can ask you.

- 1.4.9 ¶11 **Problems:**  
*..., but is a known barrier to entry for customers...*
- 1.4.9 ¶12 1Again, during the pre-work info gathering phase you should have made a problem statement. 2Again, just one. 3Sometimes this is a bit artificial, so you don't need to include it. 4The business objective might be enough. 5If there is a known problem to be solved or, as in here, a related but critical problem, include it.
- 1.4.9 ¶13 1Note, this is still a business-level issue. 2Problems that are core to business process or — as in this example — would severely impact the company if they get worse, or if they get solved.
- 1.4.9 ¶14 **User Needs:**  
*... To assure as many customers as possible complete registration or conversion, the process must be as seamless as possible:*
- 1.4.9 ¶15 1This can start, as this example does, with a customer goal as well. 2This should have come out of the pre-work above as well, but if it doesn't, or it doesn't fit, then you may modify as needed.
- 1.4.9 ¶16 1You will need to include the high-level description of the design goal. 2I say you should pick this from the morphology analysis chart at the beginning of this chapter. 3It can be re-worded if needed to make sure everyone understands it, but its still a guiding principle.
- 1.4.9 ¶17 **Specific Design Tactics:**
- *As few fields as possible should be presented*
  - *Whenever information is systematically available, it should be pre-populated*
  - *Information should be grouped by task*
  - *Labels and descriptions should be clear, truthful and use standard terminology conversion, the process must be as seamless as possible*
- 1.4.9 ¶18 1These are going to, usually, be heuristics. 2They can be expanded to include details for this product, but if it doesn't tie to something already known to be a good thing — either specific to previous variants of the product, or things that are heuristically known to be good design — then remove it or re-write it.

- 1.4.9 ¶19 1Note that these are tactics, which can be defined as “an expedient for achieving a goal; a maneuver.” 2They are not designs, but specifics that lead you to the design eventually. 3Everything else in this phrase, even the morphology has been strategic. 4You will not be able to come up with these tactical bullets without more design background. 5We’ll be talking about heuristics later on, though, so don’t worry about it just yet.
- 1.4.9 ¶20 1Once you have finalized the design objectives, stick to them. 2If you have to evade a design objective, stop and figure out why. 3It might not be bad for the project, but you cannot just skirt the design objective as you want to. 4You need to go back, change them, and apply it universally.
- 1.4.9 ¶21 1This might be hard to do, sure. 2You might not be able to get it into this release of the product for time or political reasons. 3Therefore, work hard on getting it right. 4But don’t break the objectives. 5They will keep you honest during the later design phases.
- 1.4.9 ¶22 1Speaking of which, this is design also. 2Its not drawing, but its certainly design. 3It’s an underlying task, one you really need to define the scope of work. 4Among other things, this means that you should avoid delegating it to some other team. 5You, as the designer, need to work on this, as part of the team if not all by yourself.
- 1.4.9 ¶23 1It also means that it will be public-facing. 2Make sure its reasonably well-written and understandable to everyone you think might need to see your documentation.

## 1.5 **Divining and Prioritizing Requirements**

- 1.5 ¶1 1Objectives, and the related analytics and artifacts above, are at a very high level. 2You could instead consider them vague.
- 1.5 ¶2 1Eventually, in a technical system especially, your product will have enumerated requirements. 2How these are formatted depends on the process being followed, and to a degree the whim of the requirements manager (or equivalent)
- 1.5 ¶3 1Requirements are, unfortunately, not something you will usually be allowed to make on your own. 2These will be created by technology teams and often by specialized requirements creation teams. 3Make sure you understand how this process works, who is involved and your role in it.

### 1.5.1 **Business Requirements**

- 1.5.1 ¶1 1Normally, if you just ask out of the blue, you'll be told there is no role for UX in requirement development. 2Actually, if you get calls returned you are doing pretty well. 3But this first level of requirement setting is critical to the success of the project, and is a key area where usability and design can influence that success.
- 1.5.1 ¶2 1Ideally, this process won't happen, and you will get months to perform research and write your own recommendations, which form the core of the design. 2This is good, and you should try, but it's rare in my experience and certainly outside the scope of this document. 3The design objective development process outlined above can feed into this, but often will not, so don't be disappointed when they are ignored to a certain degree.
- 1.5.1 ¶3 1Business requirements are so called because they address the needs of the business. 2In concept they focus on product and marketing, but in fact much of the set is user-focused, or can be easily approached from that point of view. 3Business requirements address the "what" of a project, but should never talk about "how."

1.5.1 ¶4 1Whenever possible, insert yourself into the process. 2This is easier than it might seem. 3While the product owner will technically “own” the requirements, and might feel this ownership deeply, it turns out it is boring to write requirements. 4No one likes doing it, and fewer still like editing them for accuracy, spelling and conciseness. 5You can volunteer to help with this, and just add your own thoughts to the process.

1.5.1 ¶5 1There should be only 5-15 business requirements. 2Anyone who insists there should be 50 or more is writing functional requirements. 3Cutting down the number of requirements and merging them also something you can help improve.

1.5.1 ¶6 1Several typical requirements you might see would read:

- *System shall have the ability to allow user to enter a street address and geocode it to a dynamic map displaying coverage and signal strength*
- *System shall have the ability to drill down from a national map perspective if no address on file for user (i.e. in-front of log-in)*
- *System shall have the ability to allow user to enter only a ZIP code and geocode it to a displayed map*
- *System shall have the ability to allow user to enter an intersection and geocode it to a displayed map*
- *System shall have the ability to allow user to enter a City/State combination and geocode it to a displayed map*
- *Etc.*

1.5.1 ¶7 1These are even written as functional requirements (system shall). 2You can consolidate them to a single requirement, covering all the needs without specifying the technical method or design:

- *Allow customers to find coverage for a specific area in multiple ways: by entering a general location, a specific street address, an intersection, two locations, etc.*

1.5.1 ¶8 1Always keep in mind your design objectives, the design morphology and any other items you

have already developed that can help you.

## 1.5.2 Functional Requirements

- 1.5.2 ¶1 1Functional requirements are more specific, and even more boring to write so you can continue to insert yourself into the process. 2They talk in great detail about the “how” of implementing a project.
- 1.5.2 ¶2 1While this will take a great deal more time, the specificity with which such requirements are typically written makes it critical that the UX team be involved at some level. 2If not, your designs will be in conflict with the requirements. 3Pretty much every time, you and your design will lose this argument.
- 1.5.2 ¶3 1You might think — or be told — that if the FR document is the final determination of what will be built, every one of your design requirements needs to be in this document. 2Not true. 3Many other documents are used to specify the final state of a project. 4To get accepted as one of these, you will need to approach the design documentation very rigorously, seriously, and with great specificity. 5But it can be accepted as a formal deliverable to development and test.
- 1.5.2 ¶4 1Hopefully, functional requirement creation is much later in the design process than is implied by its placement at this point in the book. 2You will want to do this typically just after the high level design documentation (detailed later on). 3Often, if you have communicated this sufficiently, requirements managers will just steal your design documentation and write it up as requirements, at least partially.
- 1.5.2 ¶5 1You will still need to go through these to make sure there are no technical solutions, or implied UI, perhaps in some other section even. 2Sorry, but you will need to read all of them.

## 1.5.3 Personas, Profiles and Segments

- 1.5.3 ¶1 1The two requirements documents are described at this stage in the design procedure because of their analytical value. 2The other key to using

these to add value to your design is the user.

1.5.3 ¶2 1Sure, you can approach things in a vaguely user-centered way, and eventually get good at applying heuristics based on the manner in which users behave, but building personas is perhaps the most significant, specific and useful one.

1.5.3 ¶3 1Creating personas is a deeply involved topic in itself. 2I have not one but several books that occupy themselves entirely with persona creation and use. 3But there are a few points I will make about them, and mostly about what they are not.

1.5.3 ¶3  
Persona development, and a thorough understanding of what they are and are not, really do consume thick books. The core philosophy, principles and methods are laid out in Allan Cooper's various versions of *About Face: The Essentials of Interaction Design*. Version 3 is coming out momentarily, and is probably the recommended edition to get. For over 700 pages about nothing but personas, try John Pruitt & Tamar Adlin on *The Persona Lifecycle: Keeping People in Mind Throughout Product Design*.

### 1.5.3.1 Personas

1.5.3.1 ¶1 1A persona is an archetypal user, essentially a single fake user that has the characteristics of a usefully representative portion of the expected user population for the product.

1.5.3.1 ¶2 1You should create personas for each program. 2Projects are grouped into products, products are grouped into programs. 3So while you are likely to create a persona for a project, under its timeline and budget, do it with the over-riding program (e.g. online customer care) in mind.

1.5.3.1 ¶3 1Persona creation is officially done by interviewing many actual users, over long periods and with lots of complex analysis. 2It's great. 3I have used such personas exactly once. 4No one ever has the time or budget for this. 5But all is by no means lost. 6The modern, interconnected world means that people talk about themselves all the time. 7Search for then skulk around forums, search for resumes, read blogs by representative users. 8Then follow the typical creation process by taking those attributes and consolidating them.

1.5.3.1 ¶4 1You also should not end up with one persona. 2Several will be created, as a whole encompassing the bulk of the users who are expected to use the product. 3Beware of being caught up in interesting edge cases that will influence you to make changes for small user groups.

1.5.3.1 ¶5 1Some persona processes promote the use of one primary persona for the entire program or

project. <sup>2</sup>I think this is terribly flawed, and only addressing the needs of a set of personas gives the benefits of the analytical tool. <sup>3</sup>While some will end up being primary, due to their high rate of use of the product, that cannot be determined just yet.

**1.5.3.1 ¶6** <sup>1</sup>Like the relationship between design and drawing, a persona is the fake user, and not really the document itself. <sup>2</sup>However, most consumers of the persona will get it from the documentation, so consider carefully what and how you want to share it. <sup>3</sup>A typical persona document gathers interesting tidbits of information, behaviors and attitudes and consolidates them into a single thread:

### **Sky Wojcik - Mobile Youth**

A single mother and recent college graduate, Sky has moved back in with her parents to make ends meet. Computer-savvy, she just doesn't have space or money for a computer in her personal life. Her phone is really her only personal space.

#### **Connecting**

Sky lives by texting. It's used for coordinating, checking in, and sending "I'm thinking of you" type messages. She has invested in a large text "bucket" so she can predict how much each month will cost. Despite the cost sensitivity, Sky has downloaded a different ringer for each friend. She is startled when the phone rings with the default ringer. She wants the same to be true for text messages.

Several of Sky's friends are starting families, and she has learned to dread the obligatory exchanging of cards at showers; it's so artificial. But she keeps giving them. She will send cards for a variety of occasions, and keeps all important cards (not the ones from the doctor's office). Displaying them? "Only old people do that." She doesn't notice that her phone's wallpaper might serve the same purpose.

Sky has a number of friends now in the military and deployed overseas. She exchanges hand-written letters with them. She doodles on cards to personalize them, but if she can't find the "right" card, she'll just send something close enough. Sometimes that means the emergency trip to the grocery store and hiding in the next room to sign the card.



**Technology**

Sky is part of “Generation Y.” She understands Google, texting, MMS, Wikipedia, Facebook, MySpace (“but that’s for high school”), and they affect her expectations for new services. She’s lived online for the past few years, and with a bit of digging you can find the gory details of her break-up with her boyfriend from her senior year of high school, available for the world to read.

Sky will tell you she doesn’t “browse the mobile internet,” but she frequently visits the celebrity gossip site on her phone. She also thinks that the phone is safe from any viruses.

**1.5.3.1 ¶7** 1Quotes are from various actual users in the set used to create the persona. 2These can add to the understanding by the project team that the rest of the information listed is trustworthy and true.

**1.5.3.1 ¶8** 1Another way to write up a persona is to tell an individualized (but still composite and “fake”) story:

**Janice Sheehan** has been the Telecom Manager for Ford’s Dearborn plant for the last 4 years. She manages not only the telecommunications services, but a team of 12, (8 on day shift and the rest floating variously to cover evenings and weekends). She also has an administrative assistant who helps with a lot of the reporting and financial analytics. All her employees are hourly staff, and are unionized along with most other plant employees.



The Dearborn plan is a big complex, so Janice’s job is about as much effort as that at lots of entire companies. Auto plants are really very large items, made up of a number of factories, office buildings, design centers and other facilities. This one includes the Rouge center, a general advanced technology and environmental showcase (largest living roof, paint fumes power fuel cells for electricity, etc.) where they are actually building F150s.

This is a very demanding job, with many experimental, automated systems to manage, often thru various “tele-supervision” schemes, where supervisors may be in an office on the other end of the plant, two miles away (or theoretically, at home). Knowledge of network outages, planned and unplanned, is critical to the smooth and profitable operation of this facility, and the continued success of these projects...

**1.5.3.1 ¶9** 1This is just a snippet of one. 2In this case, the specifics are not quotes and usage figures, but a

mixture of actual tasks, job functions and typical personal behaviors and attitudes of the sample users consolidated into the persona.

**1.5.3.1 ¶10** 1Of course, you can also use both of these methods together, or more. 2There are any number of other ways to share a persona with the project team. 3Create a storyboard depicting the life of a persona, make a live-action movie with narration, or act it out in team meetings. 4Some companies have even hired actors to work at the company full time and pretend to be a persona; you can go into their cube whenever you want, and ask them questions. 5Many of these can take the concept to extremes they probably should not, but they are good for us to consider as ideas.

**1.5.3.1 ¶11** 1Which one you use depends largely on the type of project. 2The storytelling style is useful for wholly new types of users, where the project team may not understand how the project will really be used. 3The user-summary style shown first is best when everyone thinks they understand basically how the product will be used, and wants to understand what sort of people will use it, and how their individual use will vary from the other personas modeled.

**1.5.3.1 ¶12** 1Other types of user definitions are often called personas, especially from about 2005 onward. 2I guess as part of the “usability is good” thing where everyone adopted buzzwords, but few adopted the principles. 3Do not assume a persona you did not create is actually one, and look out for these two poseurs:

**1.5.3.2 1.5.3.2 Segments**

**1.5.3.2 ¶1** 1Segments are marketing tools, which divide the entire user base into a series of categories (or segments). 2Within a category, demographic or other information is averaged or otherwise presented as aggregate information for the entire population being discussed.

<b>A</b>	<p><b>1.1.5 Young Connecteds</b> 26% Key Benefit = Mobility</p>	<ul style="list-style-type: none"> <li>• 20s – 30s, high income &amp; education, majority single</li> <li>• 1/3 minority (15% Hispanic, 15% African-american)</li> <li>• Heavy voice and data usage</li> <li>• Least price conscious</li> </ul>
----------	---	---

- 1.5.3.2 ¶2 1A typical market segment looks like this:
- 1.5.3.2 ¶3 1But beware, they are often stretched to look like personas, even to include life stories, names and photos. 2Always look at how anything given to you as a persona was made, and automatically be suspicious of anything created by marketing, or by vendors employed by the marketing organization.
- 1.5.3.2 ¶4 1Segmentation can be a severe pitfall due to a few key factors:
- 1 The entire user population is included in the segments. There is a tendency to try to reach consensus design for each segment. But these are not personas; there is so much variation when considering the entire user set this is not possible
  - 2 Aggregate information must be used as aggregate information. There is a tendency to assume the average is typical, and again to create personas on the fly, or assume an average customer that does not exist
  - 3 Demographics are not people. Even when past purchase decisions are taken into account, even when focus group information is (misguidedly) included to add desires, this does not connect needs to behaviors in a robust, reliable manner.
- 1.5.3.2 ¶5 1But it is still useful information. 2While you can certainly use it as some of the beginning information for persona creation, you can also take it and apply (as we'll talk about below), albeit carefully, to analyze the design features.
- 1.5.3.2 ¶6 1You can also use the most quick and dirt of the persona creation techniques to turn the most useful of them into profiles in fairly short order.

### 1.5.3.3 Profiles

- 1.5.3.3 ¶1 1Profiles are smack in between these two. 2They take the same conceptual approach as personas, by considering user- and use-centered attributes, but are essentially lists of aggregate statistical information like segments.
- 1.5.3.3 ¶2 1In creating personas, you can usually gather profiles as a mid-point working document, and

if needed as a deliverable. **2**Moving from profiles to personas takes some time, so these may be useful if your timelines are getting short.

**1.5.3.3 ¶3** **1**The key to using profiles correctly is to not overstate their importance. **2**Don't assign them names, photos or anything like that. **3**Try not to personify them, or identify them to actual users interviewed. **4**If this helps, to create an analogy to explain behavior, prevent that example from becoming an ad hoc persona for the project team.

#### **1.5.3.4 Feature Evaluation**

**1.5.3.4 ¶1** **1**These personas can then be used directly in a sort of quantitative analytical tool to evaluate the proposed project, and also to develop the personas into a more useful tool for the rest of the project.

**1.5.3.4 ¶2** **1**Start by developing a list of features or attributes of the proposed design solution. **2**These are not always real requirements. **3**They should be usability-driven, and will often be a consolidation of the business requirements and design objectives.

**1.5.3.4 ¶3** **1**If needed at the point where functional requirements already exist, some of those can also be added into the mix. **2**If documentation from previous releases is available, those functional requirements may also serve to guide you in creating typical tasks or attributes to evaluate.

**1.5.3.4 ¶4** **1**Then run all those personas through this. **2**Ideally, have a different designer or other usability professional perform this for each persona. **3**If you don't have a large enough team (and you usually won't) just explain the persona, and the project, as neutrally as possible, and accept their input. **4**There is usually no benefit to checking their work by having multiple designers do each persona; the results are always very close to each other.

**1.5.3.4 ¶5** **1**Use at least three personas, to avoid two-matching implying unanimous.

**1.5.3.4 ¶6** **1**Take the values and decide which (two?) get to be our primary personas for each category of use

	Requirement	A	B	C	D	E	U
<b>Capabilities &amp; Functions</b>							
1	Graphically display all mobile phone/data coverage types	3	4	3	3	3	4
2	Graphically display the location of Sprint towers	2	2	2	2	2	-
3	Graphically display Sprint WiFi Zone coverage areas	4	2	3	3	4	3
4	Graphically display the location of Sprint stores and other retailers selling Sprint equipment	2	3	3	4	2	3

case. 2Above is a sample one.

1.5.3.4 ¶7 1In the example above, columns A through E are personas, defined somewhere else (but not any of the examples above, sorry). 2The “U” column here consolidates the results of a paper-prototype user study to compare real people on the real product. 3If you have more rigorous user study information, the participants can be broken down to match the personas as a validation check, but this technique does work without users at all.

1.5.3.4 ¶8 1The scale I usually use is:  
 5 User’s life improved/loyalty improved/etc due to value of feature  
 4 User would gain value from this feature  
 3 User would rarely use, or would use only by default  
 2 User would be unlikely to use this feature successfully  
 1 User would actively dislike this feature

1.5.3.4 ¶9 1It may be a good idea to color code the values to make it easier to scan. 2Many I have worked with use codes in the cells for scannability, like --, -, 0, +, ++ but using numbers makes it easier to do math if that is needed.

### 1.5.5 Task Inventory

1.5.5 ¶1 1The Task Inventory is, when time provides, the

next step to perform. **2**To perform this, you will need a series of tasks. **3**Usually, these tasks are based on the functional requirements, but if the process is much more user-centered (instead of being driven by IT process) the tasks can easily be made from the design objectives and other design concepts.

### 1.5.5.1 Use Cases and Task Lists

1.5.5.1 ¶1 **1**Develop a series of tasks, in much the manner as a use case. **2**The user enters the system (with some pre-conditions), performs a series of tasks and then completes the task. **3**Several series of these should be performed. **4**This is not a complete model of the system, in any way. **4**It should not try to depict every state, and will not include options and branches.

1.5.5.1 ¶2 **1**Instead, much like persona creation, it touches on the key tasks to give a representative view of the system.

1.5.5.1 ¶3 **1**Just take the task in detail and walk through it. **2**I suggest doing these with one profile per task. **3**A second column would also be an acceptable way to annotate it, but this can influence the results. **4**When possible, grab some other designers and have each of them perform the task inventory for a single persona. **5**If there are not enough designers on the project team, just explain the project as simply as possible, give them the persona information and the task inventory and let them go.

1.5.5.1 ¶4 **1**On the next page there's a snippet of one I have done. **2**The "req" column here denotes the functional requirement number. **3**Note that any one task will represent several functional requirements. **4**This is useful for tracking, so you can prove to the project team, developers and others that your process ties to theirs, and that the results are relevant to the formal requirements.

1.5.5.1 ¶5 **1**The value column is the same as that used in the feature evaluation. **2**You may either rank the individual task, each requirement, or both of them. **3**If the design process is going well (and you have not been required to add features that failed to pass the feature evaluation process)

Step	Task	Req	Value
1	Visit Sprint.com, sign on, arrive at My PCS page	34	3
2	Probably regular visitor, so aware of personalized coverage, but has not specifically looked at it; looks at it now	34	4
3	Already centered about billing address (home address), but map is small and un-detailed; click into detail view	1 43 45 19	3 4 4 4
4	Detail view has multiple controls and affords to googlemap drag interface mode (shadowed layer for centroid marker); user explores drag and zoom interface for cool-factor	37 33 32 24 22	3 4 4 5 5
5	User checks map for trustworthiness: Finds a known bad spot and sees if map reflects this (assume satisfaction with accuracy/trust)	19 50	4 5

all the ratings should be 3, 4 or 5. <sup>4</sup>Don't limit yourself to this during the walkthrough; just keep it in mind when analyzing the values later.

1.5.5.1 ¶6 <sup>1</sup>Always perform the task inventory for the primary personas. <sup>2</sup>If possible, also do it for the secondary personas. <sup>3</sup>In this case, come up with a manner of representing that the secondary personas have less weight.

1.5.5.1 ¶7 <sup>1</sup>Yes, you can also use your favorite methods to analyze the results here. <sup>2</sup>Typically, average the results of the primary personas for each task, but if you have access to a statistician, other methods can be used as well.

### 1.5.5.2 Consider the Negatives

1.5.5.2 ¶1 <sup>1</sup>For both of these analytical methods, consider carefully what poor scores mean. <sup>2</sup>Don't just assume they are bad, and toss them out, but try to determine why they do not meet the needs of the user.

1.5.5.2 ¶2 <sup>1</sup>This is especially important if you are required to have a "bad" feature for technical or business reasons. <sup>2</sup>How much harm is it really doing? <sup>3</sup>Will that harm only be to some customers, in some situations? <sup>4</sup>Most importantly, is there a way to change it (or hide it) so that the feature works better or fails to bother users.

## Section 2

# **A Process for Design**

## 2.1 A Methodology For Information Design

### 2.1.1 So, Here's the Process?

2.1.1 ¶1 1Of course not. 2As described above, this is a methodology. 3A manner in which you can approach design. 4In fact, this is not just a choice of words, but an important distinction. 5Read the section about process above if you don't get it.

2.1.1 ¶2 1On the other hand, you should already be in the design process. 2The gathering of information, its analysis and the creation of the design objectives is part of it also. 3I claim its not possible to start this methodology in a vacuum; even if you aren't formally executing a gathering and analysis phase, you are doing it anyway. 4Maybe subconsciously, but it's happening.

### 2.1.2 When Do I Get to Draw?

2.1.2 ¶1 1The information in this book has been designed as much as any other product, and like most designs there have been some sacrifices and single-point-of-view choices made. 2One is that tossing the whole thing at you in one bundle, in the order you do it, is impossible to understand.

2.1.2 ¶2 1So, this very next part is about what you will draw, when you pick up a pencil (or mouse), then it is followed by a chapter on sketching and much later is information on how to create final design artifacts you can share with clients and implementation teams. 2In practice you will do these things in an overlapping, intertwined manner.

2.1.2 ¶3 1You should, therefore, draw as soon as possible. 2Read though the Information Design Phase portion below, then immediately use the methods following to try them out. 3Design — but most of all these design methods — has evolved to exploit the interaction between your conceptualizing brain and your drawing hand.

2.1.2 ¶4 1If in doubt, if you wonder if it's okay to draw, the answer is always “yes.”

### 2.1.3 Summary of the Information Design Phase

2.1.3 ¶1 1To ground you, here's a summary of the steps we are going to detail out shortly:

#### 2.1.3 ¶2 Filter

- 1Hide all information, process or system interaction the use will not need.
- 2Make visible all information, processes or tasks the user will want or need.
- 3Define the limits of any user interactions or customizations.

#### 2.1.3 ¶3 Group

- 1Determine user goals and tasks within the remaining information & functions.
- 2Group information and functions by task and goal.
- 3Group all other information by user logic.

#### 2.1.3 ¶4 Prioritize

- 1Organize groupings hierarchically by user interaction, user logic or task process flow.
- 2Offer the lowest number of user interactions (clicks) possible.
- 3Consider architectures that present multi-layer information in an apparently flat manner.

#### 2.1.3 ¶5 Arrange

- 1Place items on individual pages based on grouping and priority.
- 2Specify a single template for use on most or all pages.

#### 2.1.3 ¶6 Optimize

- 1Revise the design, iteratively and with user feedback if time allows, to assure it is the right solution.
- 2Confirm with technical resources that it can be built and modify as necessary if not.

### 2.1.4 Actually, First Gather

2.1.4 ¶1 1Before you can filter items, you have to have the items, and throw them up on the wall (or whenever you put them). 2A lot of this stuff you will already have from existing requirements gathering processes, from research or from the

previous version of the product. **3**This is detailed in the later parts of the previous section. **4**So if you jumped ahead to the process parts, go back and read that whole thing.

- 2.1.4 ¶2** **1**Regardless of what I said above, and the fact that I refer to these items during the design processes here, there is no particular magic formula. **2**You can gather information in essentially any manner that works for you, or is available based on the project. **3**Sometimes you will inherit half-completed work, or not be able to gather the information you need, or otherwise be constrained. **4**And, you will change over time as you pursue the process, with each project and as you develop your own style.

## **2.1.5 Logistics and Process**

- 2.1.5 ¶1** **1**The process of information design itself can be done individually, or in a group setting. **2**A whiteboard is a good way to lay out the information, particularly at the start. **3**Cards or post-its also can work well, often on the whiteboard so other notes can be made. **4**Methods are all discussed in great detail in the next section Sketching Towards Understanding.
- 2.1.5 ¶2** **1**The information design phase is pretty quick, and should be done, if possible, in a single session. **2**It might take an hour, or could take four if it's a big product and there are lots of details and arguments. **3**A couple days is fine if you count the refinements in last steps, but it's not something that takes weeks.
- 2.1.5 ¶3** **1**Each step is a guide on the way to design and understanding. **2**They do not have to be strictly completed before the next is done, though with large groups or when starting out, it is best to stick close to the procedure, and declare when moving to the next step.
- 2.1.5 ¶4** **1**If performing this with a group, and especially with a large and diverse group that includes the entire project team, consider getting a moderator. **2**Anyone can do if they have an understanding of the process, and can command the attention of the room, such as another designer. **3**This leaves you, as the primary designer, free

to assist, and to contribute your own thoughts without overwhelming others

2.1.5 ¶5 1The skills used by a moderator come from research, so if you have such a department see if you can borrow someone who is good at focus groups and the like.

## 2.1.6 Filter

2.1.6 ¶1 1To start, maybe even before everyone else arrives, place all the items that are needed at the top of the board. 2I suggest a in a single row, so no hierarchy of any sort is implied, but other methods like a pool of items on a board to the side can also work well. 3Then proceed through the steps in order, changing and moving items as specified.

2.1.6 ¶2 In the filter step, we will:

- 1 Hide all information, process or system interaction the use will not need.
- 2 Make visible all information, processes or tasks the user will want or need.
- 3 Define the limits of any user interactions or customizations.

2.1.6 ¶3 1You will, then, move each card (or mark on the whiteboard) to regions denoting visible or hidden. 2Hidden items are actually removed from the interface, but they are called “hidden” because they are often features that already exist in the system or data store, and are just being hidden from view in this case. 3Politically, this helps a lot when selling the concept to product and technology owners.

2.1.6 ¶4 1The third item, limits of interaction, can be confusing. 2This allows the design to account for items to be within the design, but of notably lower priority. 3Disclaimers are a good example of something you would try to get into this bucket; the legal department might assume that the full disclaimer is on each page, but you should be able to argue that it will be read more successfully by those who need it with a small, clear, link to the information. 4This link is the limiting feature of the item. 5While it still exists, it is pushed off to a different level of priority;

this should be noted by a border on the card, a small item saying “link” in the visible list, and a line to the side for the actual disclaimer, or similar annotation.

2.1.6 ¶5 1If you explicitly followed the task inventory process above, ranking information is going to be available for each feature you place on the board. 2For a first pass I usually ignore these values, as everyone has an innate sense of what is important anyway, or has been paying attention to the previous parts of the process.

### 2.1.7 Group

2.1.7 ¶1 1Grouping can easily merge with filtering, and probably should as you begin to understand the process. 2But it is worth breaking out as an independent step so the decisions made as a part of it are understood.

2.1.7 ¶2 1One reason grouping is often integral to filtering, is a key portion of step 2 here.

- 1 Determine user goals and tasks within the remaining information & functions.
- 2 Use the design objectives, design morphology, feature evaluation, and task inventory to determine which tasks are used coincidentally with others for the primary personas.
- 3 Group information and functions by task and goal.
- 4 Group all other information by user logic.

2.1.7 ¶3 1Consider the task inventory. 2When creating individual tasks, or flowing from one to the other, note how tasks naturally coalesce. 3To determine which of these tasks should go together for the design of a common system, used by everyone, you have to analyze all of the task inventories (and other analytical tools) in aggregate; do not follow only one persona, and especially do not follow anecdote, preference or the behavior of a “typical user” who is just someone on the project team. 4Use your data.

2.1.7 ¶4 1Items will be grouped not just conceptually, but physically. 2See the next section on sketching techniques, but a typical one is to move the cards into groups, then draw lines around them

with whiteboard markers. **3**Many other techniques also work fine. **4**Much of this depends on your time, the artistic skills available, and how everyone prefers to work.

- 2.1.7 ¶5** **1**Note that a single session does not mean breaks cannot be taken. **2**Allow everyone to get up and stretch. **3**Sometimes, the designer can be drawing revisions, or grouped cards for the next step, at these break points. **4**The designer can then take a break while the moderator begins explaining the next phase to the project team. **5**This is another advantage of using a separate moderator.

## **2.1.8 Prioritize**

- 2.1.8 ¶1** **1**The next step in designing is to assign a priority to each element in the design. **2**Remember that at this point there is no explicit determination of page, screen or state layout; everything is considered as a single grouping.
- 1** Organize groupings hierarchically by user interaction, user logic or task process flow.
  - 2** Use the feature evaluation, and especially the task inventory to determine which of the grouped modules are the most important.
  - 3** Offer the lowest number of user interactions (clicks) possible.
  - 4** Consider architectures that present multi-layer information in an apparently flat manner.

- 2.1.8 ¶2** **1**Avoid drawing boxes and arranging items, as that is a later step. **2**Use a bullet-list outline format, or simply arrange cards or post-its in an order (top to bottom, usually).

- 2.1.8 ¶3** **1**The results of the priority list may begin to reflect a page-level view of the final design. **2**Use some method to display the elements hierarchically, instead of just listing them in a forced-rank manner.

## **2.1.9 Re-Filter**

- 2.1.9 ¶1** **1**When reading about how grouping is related to filtering, you might have wondered why I did not simply say they can go back and forth (it-

eratively, perhaps) to work hand in hand. <sup>2</sup>That is because a small amount of distance actually helps with evaluating whether grouped items should be filtered out, or previously-eliminated items can be filtered back in.

- 1) Look again at everything visible now. The lowest priority items on the list may not be needed, or may not be needed on this screen.

2.1.9 ¶2 <sup>1</sup>Follow the procedure already described under Filter above to perform this step.

### 2.1.10 **Arrange**

2.1.10 ¶1 Once re-filtering has defined a more final set of available elements, arrange the elements in a box model or similar design model.

- 1) Place items on individual pages, screens or states based on grouping and priority.
- 2) Specify a single template for use on most or all pages.
- 3) Arrange items by design heirarchy

2.1.10 ¶2 <sup>1</sup>Pages should begin to fall out from the grouping and priority steps above. <sup>2</sup>Generally, you will be able to see this immediately. <sup>3</sup>Do consider what the process is telling you, though. <sup>4</sup>Do not assume a default arrangement based on the screens from a previous release or a competing product. <sup>5</sup>It may, however, be useful to consult conceptual designs you created earlier when selling the project (see Concept of Design later).

2.1.10 ¶3 <sup>6</sup>Some elements will be available on all pages. Set these aside as special cases and refer to their meta-grouping with a new designation or card.

2.1.10 ¶4 <sup>7</sup>When creating page templates, or arranging items on an individual page, follow heuristics, standards, styles of design that already exist, and universal hierarchies of visual communication:

Position > Size > Shape > Contrast > Color > Form

2.1.10 ¶5 <sup>1</sup>These are discussed in much more detail later.

2.1.10 ¶6 <sup>1</sup>Consider timing and interactivity as well. <sup>2</sup>Use arrows, state-changing icons or cards, or other

items to denote interactions. **3**Avoid falling into the trap of screen-by-screen, fixed design.

### 2.1.11 Optimize

**2.1.11 ¶1** **1**Optimizing is really a catch-all to correct anything that you skipped, missed or are worried about. **2**But also be sure to look over all of the elements and make sure they are in the right format and at a useful fidelity.

- 1** Arrange all elements on the page as accurately as possible within the confines of the diagram fidelity.
- 2** Build or arrange to have built any coded proof-of-concept pages.
- 3** Confirm with technical teams that what has been designed uses standard components or can be built within the confines of their capabilities, time and budget.

**2.1.11 ¶2** **1**This phase will also continue after any meeting has concluded, as you turn the working documents into a more permanent artifact, first a record of the design, then a high level wireframe.

**2.1.11 ¶3** **1**The distributable artifact should be actually distributed, to assure everyone believes this is what they agreed upon, and as shown in the list above, that it can be actually built. **2**You are likely to have to walk technical teams through the design to assure they evaluate and understand it. **3**If prototypes are to be built, this is an appropriate stage to build them. **4**Be careful that they do not over-step their bounds, and stick to the design arrived at by the process.

### 2.1.12 Process vs. Artifact – Designing within UCD

**2.1.12 ¶1** **1**It is important to keep in mind that what you produce is an artifact. **2**Although it is the result of the design process, it is not the process itself. **3**The artifacts developed from this process have been devolved — and modified as needed — specifically to not get in the way of good design. **4**For this reason some of the guidelines outlined here are somewhat vague; it's up to you, the individual designer to apply them correctly.

**2.1.12 ¶1**  
UCD is pushed a lot, and reasonably clearly, by the Usability Professionals Association which you can find here: [www.usabilityprofessionals.org/usability\\_resources/about\\_usability/what\\_is\\_ucd.html](http://www.usabilityprofessionals.org/usability_resources/about_usability/what_is_ucd.html)  
They have a poster that is particularly nice as an intro and reminder of the process, but it seems to be only available on paper, so I can't send a link to it to look at, or embed one in this document.

2.1.12

There is even an ISO standard (13407) for UCD, or Human Centered Design, which I have yet to read as it looks long, is \$100 to get just the PDF, and sounds boring. I hear from others that it's as scary and boring and excessively codified as I would think, so is no fun to use for design at all.

2.1.12 ¶2 1Additionally, the wireframe is supposed to actually encourage good user centered design. 2The key to this is the visibility of the elements and the robust description of the overall system from the user's point of view. 3None of this should be new to you. 4I've been expounding on the value of drawing (or artifact creation) in adding value to design, all along.

2.1.12 ¶3 1Checks can be done for overall simplification, the completeness of error states and the mapping between cause and effect by viewing individual elements and following the flow lines. 2All processes must have consequences and results, and drawing flow arrows reveals these. 3If something is missing, it rapidly becomes obvious, simply because it does not fit into the rest of the diagram (i.e. flow lines will go the wrong way or will cross groupings).

2.1.12 ¶4 1The use of the wireframe, and its lack of use, has been observed closely over the several years of its existence. 2Like many successfully processes, its hard to prove good value. 3However, a small number of projects where a wireframe was not made have resulted in failures from missed alternatives and states; these have resulted in late-process changes which should not have occurred.

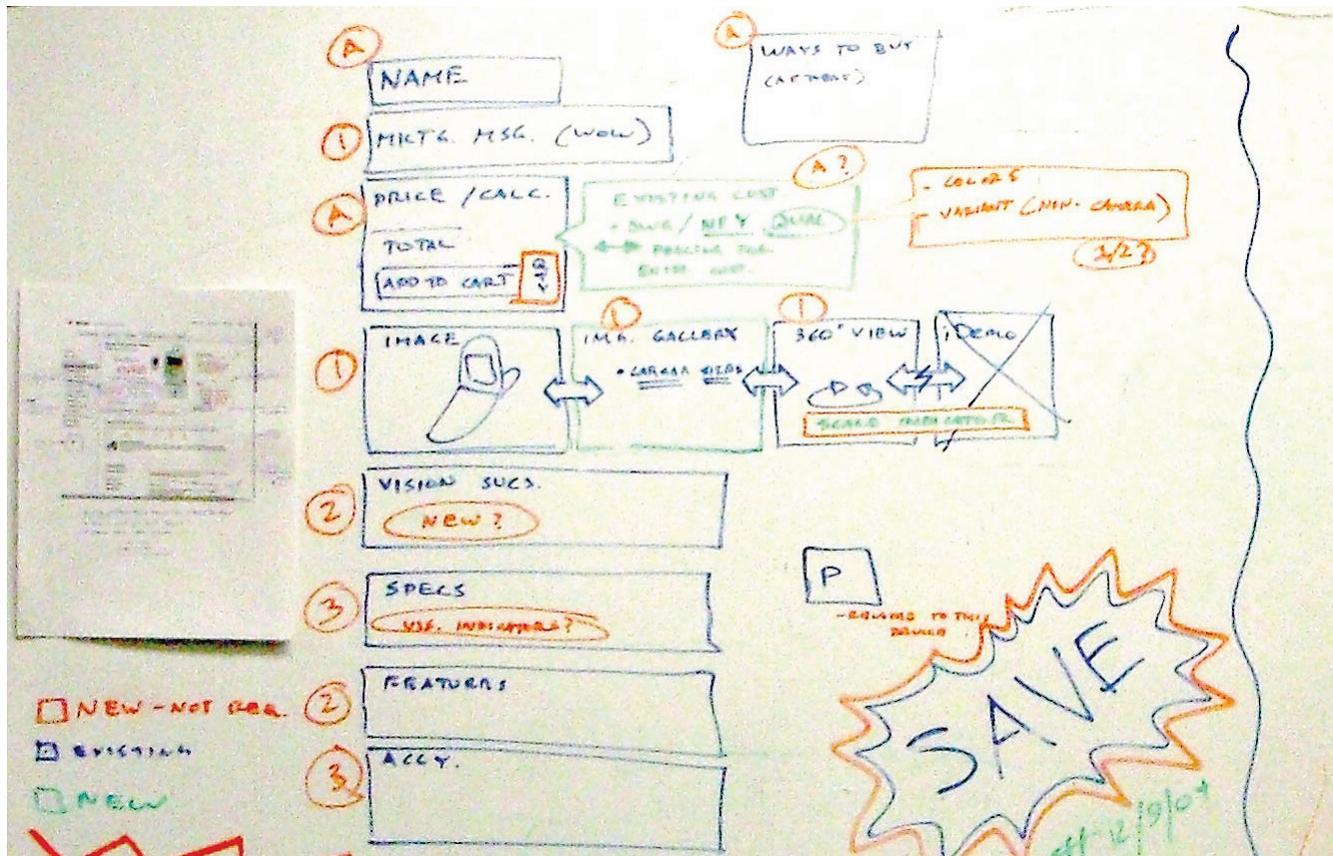
## 2.2 Sketching Towards Understanding

2.2 ¶1 1As I have been mentioning, repeatedly, to preserve and understand the information you are gathering in the information design process, you have to put it on paper. 2Or whiteboard. 3Or post-it, or maybe even a computer.

2.2 ¶2 1The next major section discusses artifact creation, so this chapter would seem to be out of place, as one that discusses drawing. 2But there is a difference between what we'll call sketching vs. drawing. 3Any process where you get your thoughts down on paper, especially collaborative processes, are covered here. 4Later, you need to create more formal, deliverable drawings.

### 2.2.1 Box Models

2.2.1 ¶1 1Box models are about more or less pure infor-



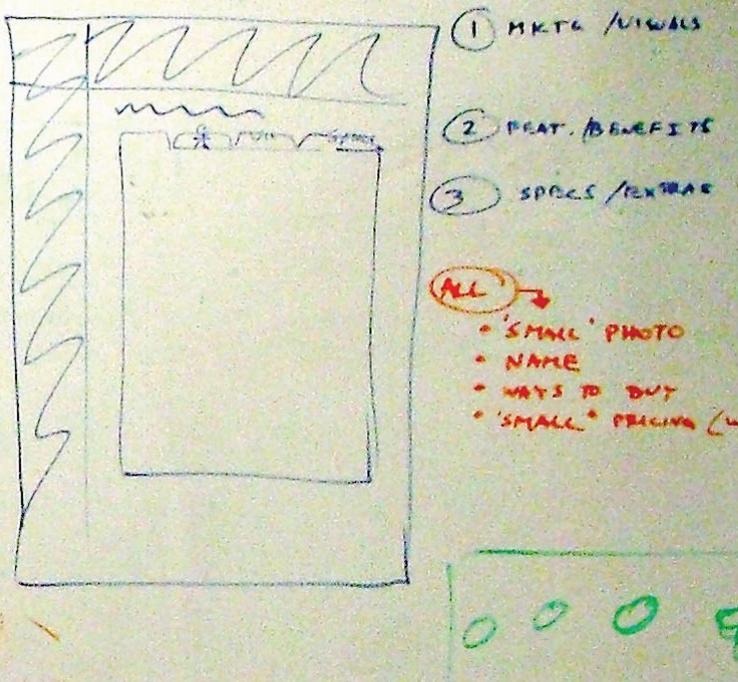
2.2.2 ¶6 et. al.  
 Many artifacts shown here are real items from real projects. They are all old enough not to matter, or have been sanitized to hide any secret information. If you represent the product, and disagree, tell me why with some specificity.

mation design. 2No interaction is shown, but it may be implied. 3It's the closest manifestation of the information design process above, where you gather, group, filter, and so on. 4You also get to, in most of these, decide approximately where each element will go on the finished screen, and how many screens you need. 5Sometimes boxes will merge, disappear, split, etc. 6Let it happen. 7Do keep in mind grids already established in design principles. 8Five elements cannot be fit across a 4-column space, in general, so don't let the basic layout be rejected by graphic. 9design or technical teams.

2.2.1 ¶12

1Usually, this is best done as a team exercise. 2Not necessarily with the clients or anyone outside the design team, though it can be used well for that sort of collaborative or participatory design. 3Especially exercises like the post-it can be done in an hour flat for reasonably complex products, so try to grab some other designers

2.2.2 ¶6 A product sales screen designed in box model on a whiteboard



from your team. **4**Get a quiet place, with no email, no phones, etc. **5**Have all the markers, or be able to project or whatever will be needed. **6**Try, as the lead designer, to prepare in advance.

**2.2.1 ¶3** **1**As you, and others, proceed to detailing the design, you need to keep in mind these relationships you have established. **2**Sometimes, changes will cascade and you will need to move elements. **3**While its also okay to come back to this step to do that, usually you won't have the time. **4**So, its important to note these relationships for later reference. **5**Some sort of annotation, or box grouping or something should accompany the drawing. **6**As these decisions are often going pretty fast, a note taker separate from the person

**2.2.1 ¶4** **1**If you have a large, complex product, there may need to be multiple sessions to design different screens. **2**Typically, there is one core screen that is a challenge when pursuing exercises like this, so I have almost never done more than one blue-box diagramming session for a project. **3**It can be run for several screens at once. **4**Especially for information-heavy designs, or where the overall information architecture is not settled, elements may move from one screen to another, or troublesome elements discarded from the main screen may coalesce into another one. **5**Plan accordingly so you have room for such eventualities

**2.2.1 ¶4** **1**These three processes are presented in a reasonable order. **2**You could create a list, then do a post-it exercise, then finish with a blue box diagram. **3**In practice, you probably will not need all of these, or you will not have time. **4**You can also mix them up. **5**Deep into detailed design, you can come back and do any of these to make sure the project is on track or to solve a design problem.

## **2.2.2 Lists**

**2.2.2 ¶1** **1**For this, I usually try to break out a word processor. **2**Color coding, bullet points and other such elements are helpful, so something like MS Word is the right way to go here. **3**If doing this as a team exercise, be sure to project; don't rely on reading things, so make sure everyone can

## 2.2.2 ¶3 Moderating a design session with a list



### Objectives: (from Tracey, Eason and Vai are in agreement)

- Lowest cost self-service channel the most effective, efficient and preferred channel.
- Drive self-service revenue (sales, transactions, and resolutions)
- Use search tool to allow customers direct access to info they desire which is relevant to them and leveraging a multitude of repositories (internal, 3rd party websites - Manufacturer)
- Facilitate self-service by helping customer move further down path in self-service - anticipate their next steps
  - Merchandise search results when appropriate with in context messaging
  - Deflect email via effective pre-emptive search functionality

Problem: Three different areas of content depending on billing system - will be worked outside of Phase II, need a fix quickly - possibly personalized support.

Customer cookies is not a viable option.

Phase II Requirements were written to be very flexible, defer to UXD recommendations.

Problem will hopefully go away early 2008, when all customers are migrated.

How Do It (Quicken/Microsoft Office example - sidebar functionality)

Next Steps:

see at the zoom level you have.

- 2.2.2 ¶2 1This can be done by you alone, or by several team members individually, before gathering for one of the other methods outlined below. 2Its probably the best of the methods to do alone, so keep it in mind for long airplane rides back from the client site, or days before you can get all the designers together.
- 2.2.2 ¶3 1The photo on the previous spread shows how it can be done with a group. 2This might be the best method when you are surprised by design (if a requirements meeting ends early and they all turn to you) or for certain very complex feature sets. 3Note that the designer or a moderator is at the screen, controlling the meeting, and another team member is typing and otherwise controlling the document. 4It is difficult to control a meeting while sitting behind a computer. 5Likewise, this is difficult to do remotely, with screen-sharing software, so consider other methods of remote collaboration.
- 2.2.2 ¶4 1The overall concept is to take detailed elements and get them down on paper. 2At this first stage, many will be repetitive, or not valuable, or difficult to understand. 3That's the point. 4Just throw everything down and work on it.
- 2.2.2 ¶5 1It is also possible to do this exercise on a whiteboard. 2Here it's more obvious how it's a "box" model; though you can do it with just lists of words, I start to bring in some elements of the Post-it process (see below) and make every element a boxed item.
- 2.2.2 ¶6 1In the case shown earlier (under the Box Model section), the team came up with a list of elements needed (as it's a modification of an existing product, a printout of that page is on the board at the far left, and in front of each participant). 2Then, those items were roughly prioritized and placed on the whiteboard as shown.
- 2.2.2 ¶7 1Note the central row (with the handset drawing) is again showing dynamic interaction or conditional behavior, and that each element is also assigned another value as to type of content. 2The legend for this is on the far right. 3Although the boxes might look like they are

depicting arrangement, they are not; this is just priority.

**2.2.2 ¶8** **1Before:** You should have kept all the post-its that were not used in the requirements gathering session. **2**Grab those, and every email the client sent with an idea, and everything users said in their needs research if you have it. **3**Also, analyze all competing products, or the existing product if this is a revision, and write down (not draw, but write down) each element. **4**Descriptions should be short, in the realm of the requirements gathering sessions. **5**If it can't fit on a post-it, its too complex and you need to break it up into individual components.

**2.2.2 ¶9** **Steps:**

For digital presentation:

- 1 List each item, in no particular order. Usually, the way to you entered them works fine.
- 2 Be sure you (or your scribe) is familiar with Word, or whatever text tool is to be used for this. Don't be confused by how the bullets are working, or how to change color.
- 3 Use colors, styles, sections, or even other open documents to indicate and arrange the items.
- 4 Highlight and read out loud whichever item is being discussed; make sure everyone is on the same page.
- 5 Discuss each item in turn. This is an advantage of the list, and the digital presentation, in that not everyone is all over the place. With certain large information sets, or large groups, it can be better than the post-it styles.
- 6 Lists work for every step until you get to Arrange in the information design process above. Save that for later. But, by eliminating it, you can focus on the purity of the elements.

**2.2.2 ¶10** **1Output:** The final result will be a text document you can share or print out. **2**It should have groupings labeled in some way, so its clear what the groups are about. **3**Functions may begin to be implied by stretching definitions. **4**Discarded elements may be included as an appendix, to

show your work, and should be kept somewhere in case it comes back up. 5Ideally, there will be notes about why items were discarded, but this is often difficult to specify.

### 2.2.3 Post-Its®

2.2.3 ¶1 1I will refer to the elements here as post-its, or cards, but anything will do. 2And yes, I know Post-it is a 3M trademark. 3The important part is that they are movable, stickable and removable. 4You can also do this flat on a table, but there are serious advantages to being able to stick things even then. 5Pins would also work, as long as you have a cork board. 6Play around with your own ideas and look through office supply catalogs; there are lots of neat products.

2.2.3 ¶2 1Though I tend to force everyone to work in a very rough, removed mode, and hand write or draw each card, its possible to use higher-fidelity items. 2Printouts of existing designs, of the current product, or even of competitor products can be used. 3Just make a lot of printouts, and cut them apart into individual elements. 4Glue, tape, etc. will let you post these. 5Provide spares, and scissors for doing this during the session as well. 6And, make sure there are blanks around also. 7New or different elements will come up also.

2.2.3 ¶3 1Note that some elements can be fixed, like the map interface shown in the sample to the right. 2It's definitely going to be there, and is inconveniently too big to make on stickable paper, so use it as an anchor point. 3While not true of every design, its a good sample of how you can manipulate the process to your needs. 4There's no points for purity-of-process, so make it work for you.

2.2.3 ¶4 1**Before:** Make sure the room has a place to stick items, and enough markers and postits or other cards, for everyone involved to write their ideas.

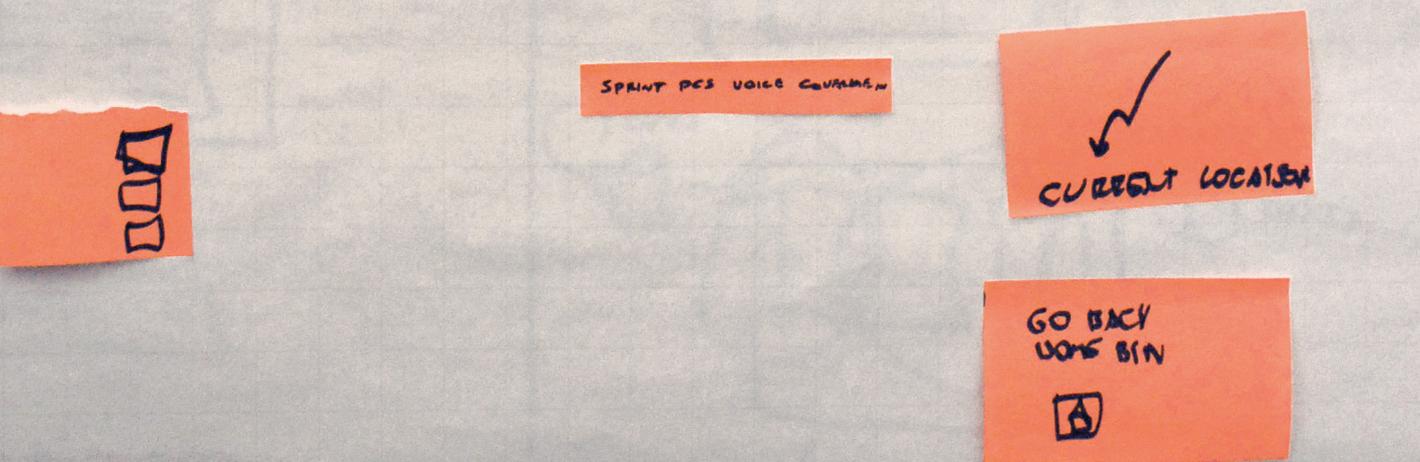
#### 2.2.3 ¶5 **Steps:**

- 1 Every time you come up with an element, draw it, or write it on a postit.
- 2 Encourage others to draw and modify elements for you. A good way to involve people is to directly involve them; if someone on

TABS AS MAP AND COUSING HIERARCHY?



2.2.3 ¶3 A map interface design using markers and Post-Its®



the project team has an idea, let them draw it.

- 3 If you have some idea what size it will be, feel free to pick the appropriate sized postit.
- 4 Provide scissors to cut down larger ones, or help with splitting concepts.
- 5 In general, simply writing does tend to imply the side of an element, even without drawing them.

2.2.3 ¶6 1Put them all up on the board. 2Follow the process above to group, prioritize and so on. 3See the finished diagram below.

2.2.3 ¶7 1Note that several elements are at the bottom of the paper, away from the diagram proper. 2Others have been discarded entirely, but these are items worth remembering, but which have no place in the design as completed.

2.2.3 ¶8 1Also note how there is other writing on the diagram. 2Not just the map, as discussed above, but a note on contextual use of tabs, some sample pointers. 3These are not elements that fit into the diagramming model, but since they came up in the meeting, they were captured. 4These might have fit best in a separate notes page, but for this project we did not have such a note-taker so everything is on the board.

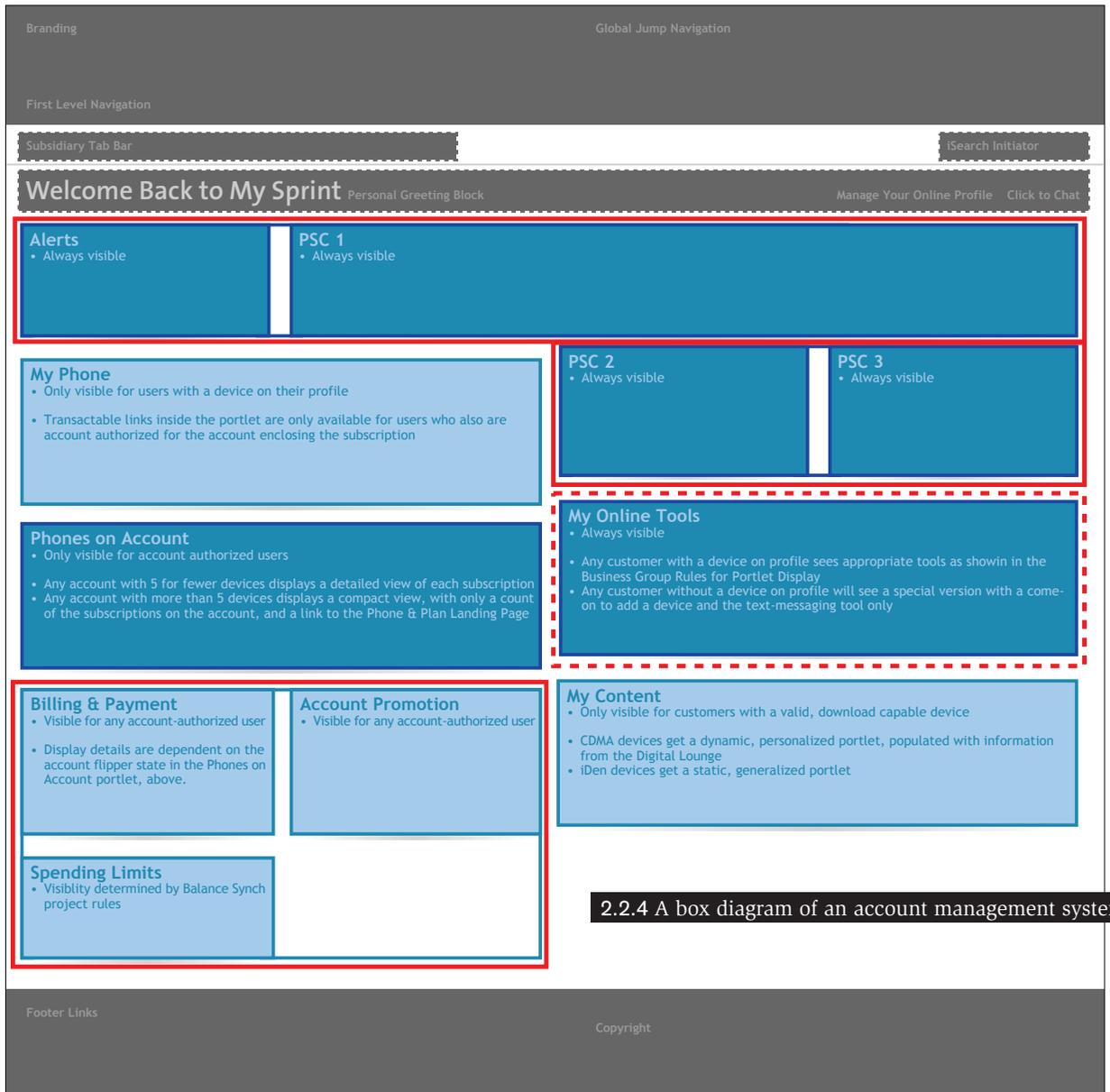
2.2.3 ¶9 1And lastly, the bracketed box to the right. It's an implied interaction. 2The function was needed, and some interaction could be accounted for based on best practices and user feedback (this is a 2.0 design of an existing product). 3However, note that it still just says "comes from somewhere." 4There is no indication of how it appears (drawer, layer, popup) or how it's activated exactly. 5That is still appropriate for this level of diagramming.

2.2.3 ¶10 1**Output:** Take a photo when done. 2Might also be good to photograph at other points along the way so you can recall the process later. 3Usually, you will turn this into a blue box diagram at your desk, but it can stand on its own. 4If on chart paper, take that with you back to your cubicle.

## 2.2.4 Blue Box

- 2.2.4 ¶1 1This is always drawn in Visio, Powerpoint, or whatever your favorite design tool is. 2I use Macromedia (okay, Adobe) Freehand, as I can then directly convert the results to my final diagramming style, but lots of people do not use it, so are baffled by that choice (and presumably I'll have to change someday as it's available-but-discontinued). 3Regardless, this is an electronic format, so content is readable and the results are permanent.
- 2.2.4 ¶2 1This is harder to do as a team. 2Not impossible, but using the computer tends to make you want everything to line up well, and generally it takes too long to move stuff. 3Also, the relatively low resolution of projected computers means not that many readable items can be on the screen at once. 4You will spend time zooming and panning. 5This is most valuable as a capture or output method once a post-it (or other) method is completed. 6But, if you have to work alone, or at least need to revise designs alone, it can be very valuable. 7The permanence and legibility can also be a bonus, for team design exercises.
- 2.2.4 ¶3 1As far as the name, these came to be called blue boxes (not originally by me) as I always used boxes of varying blue colors. 2Blue, because its never been a brand color at Sprint (that's an AT&T color). 3Therefore, its clearly a mockup or model, not a real design.
- 2.2.4 ¶4 1I don't use blue now very much. 2Just use the right color for your organization, or just make them black and white if that makes you happier. 3Make sure to use color and type contrast that will print well in black & white.
- 2.2.4 ¶5 1Resist the urge to draw, insert photos or screenshots, etc. 2Even the logo should be eliminated, abstracted or at most a faded-out grayscale. 3Except for the box shapes, just fill the boxes with words. 4Bullet lists work best for me. 5As shown above, if your site has a solid framework or wrapper, you can use it, but try to gray our outline it so it doesn't overwhelm the boxes. 6Some elements, like buttons, can be drawn, but use caution with even forms to avoid any implication of detailed design layout.

- 2.2.4 ¶6 **1Before:** Create your boxes. **2Probably,** by grabbing elements from a previous method. **3Make** them all the same size, shape and color. **4Even if** elements that in final design would not be boxed in, or your site graphic design doesn't use boxes, box in everything.
- 2.2.4 ¶7 **1Also** create a page. **2I** always work at scale, on a pre-determined typical web page or mobile phone screen size. **3Margins** are set, and titles



2.2.4 A box diagram of an account management system

of the boxes are at reasonable title sizes for the element importance.

#### 2.2.4 ¶8 **Steps:**

- 1 Okay, really you can cheat a little while making the boxes, and turn your in-head design into the box that fits your concept; just don't paint yourself into a corner, and be able to consider all options
- 2 Each box will have a very short description of its contents, and its behavior. I like bullet lists, as they don't (usually) imply a design sense, but a simpler communication of information, which is good for you at this step.
- 3 Denote conditional boxes, like those for certain customers only, differently; I use dashed lines and a tinted background
- 4 Arrange boxes using size and position. Color, contrast and shape should not be attempted at this time; make the design work with just these basics
- 5 Conditionality of display is an information design task, so should be noted.
- 6 On the diagram above, I have differentiated this with color changes (light vs. dark blue) and with line shapes (solid vs. dotted).

2.2.4 ¶9 **1Output:** A very short document will be produced which should be able to be circulated. **2**The clearly high-level output means it cannot be confused as being a final design, so if desired clients and technical teams may be given the output. **3**To tie the design to the rest of the process, include your design objectives in this document.

2.2.4 ¶10 **1**This should have been drawn in the same program you will use to generate final, detailed wireframes, so you can move very rapidly from this box diagram, to the high level wireframe, to the detailed wireframe.

## 2.2.5 Whiteboard

- 2.2.5 ¶1 1I try not use a whiteboard for these sorts of exercises. 2The main reason is that you cannot cut and paste items on a whiteboard. 3Use postits if you want that speed. 4Note that the postit method works great on a whiteboard. 5They stick well, and base elements or relationships can be noted easily in the margins. 6Of course, they are usually bolted to walls, so plan on bringing a camera instead of hoping no one erases it. 7Additionally, wording is important, and hand writing everything can be both illegible to others, and encourages less writing than when typing or writing off the wall.
- 2.2.5 ¶2 1Yes, I know there is a whiteboard example above. 2So, you can see it works okay sometimes, and they are so enticing that it's easy to turn brainstorming into a real design exercise. 3So, be aware of how to draw and capture correctly with them. 4If you want to do this on purpose:
- 2.2.5 ¶3 1**Before:** Have plenty of room, and try to get a dedicated space. 2I mean room on the whiteboard, mostly. 3If someone has a drawing up there, ask them to capture and remove it. 4Clean the board, make sure you have plenty of markers, in various colors, and bring any related supplies if you want to use post-it style basic elements. 5If it's all drawing, plan how you want to start.
- 2.2.5 ¶4 1**Steps:** Whiteboards are collaboration tools; so, this will assume you are using it with a project team. 2You will have to decide how to draw. 3Often, you as the designer will be the only one actually drawing in this case; the team will need focus, and there is often only room for one. 4It is also possible to switch off designers, or even engage non-designers from the rest of the project. 5Color can be very useful. 6Just as on electronic artifacts, use different colors for different types of elements, to differentiate content from boxes from flow, etc. 7Avoid too much black. It just makes everyone sad.
- 2.2.5 ¶5 1**Output:** Photograph the output, or transition to another diagramming method directly. 2Usually, it is still worth photographing at various steps to

document other possibilities and changes considered (or if any other artifact is lost).

- 2.2.5 ¶6 1Clearly, each of these methods have their advantages for certain types of information design activities. 2If you have a day dedicated to a design, it's plausible to use all of them as you move from typing (or writing) lists, to sketching diagrams on the whiteboard, to replacing boxes with post-its on the same board, and finishing by computer diagramming (via projector) for everyone to confirm that the transcription is true and still makes sense.
- 2.2.5 ¶7 1I have in fact done this. 2Using the right method, and switching often, can be very powerful and help everyone mentally shift gears despite otherwise being on the same task, locked in a room all day.

## 2.3 Formal Modeling Methods

### 2.3.1 Modeling Languages

2.3.1 ¶2

Sadly, I have no single reference for the design of maps. Perhaps I am too close to it, but it largely seems to be driven by internal process of a few organizations, so there is no real textbook I am aware of.

2.3.1 ¶1

1The box model methods described above form a visual language of sorts. 2A way to depict a reality in a simplified, regularized manner that others can readily understand. 3Musical or mathematical notations are examples of languages using at least some graphical notation to depict ordered information.

2.3.1 ¶2

1Maps are not photographs; they do not simply depict the ground truth, but provide a representation of this that is more readily consumable and more easily readable than a simple aerial photo.



2.3.1 A Soviet-era map depicting vegetation in great detail

2.3.1 ¶4 s1

I'd like to point out that knowledge of semiotics is entirely quantum. Those who studied it in school are entirely familiar with it, and everyone else has not even heard the word. If you already have a background in it, I apologize for mentioning it so late, and so slightly.

2.3.1 ¶3

1Such information is a language, and as such often must be learned. 2Not all symbols are innately obvious, such as the difference between the circles, and circles with tails around the solid green areas above. 3Legends try to solving this, but often poorly. 4The manner of marking maps, of creating and placing symbols, follows a very ordered set of rules in an attempt to make them as clear as possible.

2.3.1 ¶4

1Semiotics is a formal analysis of such mark-making, explicitly taught in many design schools. 2It is exposed in simple things — like hot water faucets being marked in red, and always on the left — and can be applied to any symbol creation and mark-making, especially in

## 2.3.1 ¶4

The core discussion of semiotics, as I am aware of it, is set forth in Jaques Bertin's 1967 *The Semiology of Graphics*. Serge Bonin stated, about this "graphics is a set of signs that allow you to transcribe the existing relations of difference, order or proportionality amongst qualitative or quantitative data." A good, and free, place to read a lot about this is a misleadingly titled *Semiotics for Beginners* at [www.aber.ac.uk/media/Documents/S4B/sem12.html](http://www.aber.ac.uk/media/Documents/S4B/sem12.html). While perfectly good and readable, it's quite long and involved. I am happily no more than a novice in this field.

## 2.3.1 ¶3

Fredric Jameson argues "all perceptual systems are already languages in their own right" in his 1972 book on semiotics.

a design sense, where others must understand the marks or objects, in order to use them.

## 2.3.1 ¶5

1While spoken language is descriptive of concepts, and visual language is descriptive of items (to me, I didn't look these up), modeling languages are adept at describing systems and behaviors. 2Rooted in information design, they are concerned with depicting other things or truths (information) that already exist, as with maps, or for designs, things that will exist.

## 2.3.1 ¶6

1As interactive designers, in addition to this we are deeply concerned with how the system behaves changes over time, and how any users may perceive and interact with the system through these changes.

## 2.3.2

## States & Time

## 2.3.2 ¶1

1Interactive design is not quite like many other types of design, in that users will experience only one view of the system, and one view of the information within it at a time. 2While multiple users mean that any number of views are presented simultaneously, this is usually not important from a design perspective, and only a single user's path needs to be considered at a time.

## 2.3.2 ¶2

1This view is, in systematic terms, called a state and I rather like that term also. 2It's something of a universal, with physics and chemistry referring to states as a sense of condition. 3It may be apparently stable, ordered and permanent, but is in fact transient and can change at any time.

## 2.3.2 ¶3

1Don't go overboard on this whole concept that design for interaction — in the sense of design for electronic systems — is totally unique. 2Viewing of a book can also be perceived as an interactive exercise, and a state-based experience that takes place over time as the user flips page to page. 3This conceptual relationship is useful when considering traditional designs or design methods, or even when hiring new designers.

## 2.3.2 ¶4

1But to get back on track, interactive design (as we normally address it) is inherently bidirectional. 2The system presents information in individual states, and at specific times. 3And

## 2.3.4 ¶4 s4

Semiotics and static systems: The Wikipedia article on Bertin (above) summarizes this position, that he “restricts the field of semiology of graphics, excluding musical notation, language and mathematics, systems ‘bound to the temporal linearity.’”

here is where we might begin to stretch existing design models. 4Semiology, for example, is concerned with the creation of marks in a space, but expressly not for systems or performances or other things that are designed to change or express themselves differently over time.

2.3.2.1 **Drawing states:**

2.3.2.1 ¶1 1The challenge then is how to draw interactions, versus layouts, interfaces or other static designs. 2When I first started defining this process, and after I had successfully used the information design process above, the web was still very static. 3A deck of cards mentality could be used to good effect.

2.3.2.1 ¶2 1Now, most interactive systems, regardless of their technology, are significantly interactive. 2They are applications, with numerous states and variable display modes. 3A “page” becomes a meaningless definition for most design purposes.

2.3.2.1 ¶3 1Many designers insist that prototyping is the only way to get this done. 2I have expressed other reasons I don’t like it above, but here is a key one: design documentation should help show the overall system, context and manner of use. 3A prototype can only show the actual interface, and perhaps the interaction, at any one moment.

2.3.2.1 ¶4 1A diagramming language is needed to solve this.

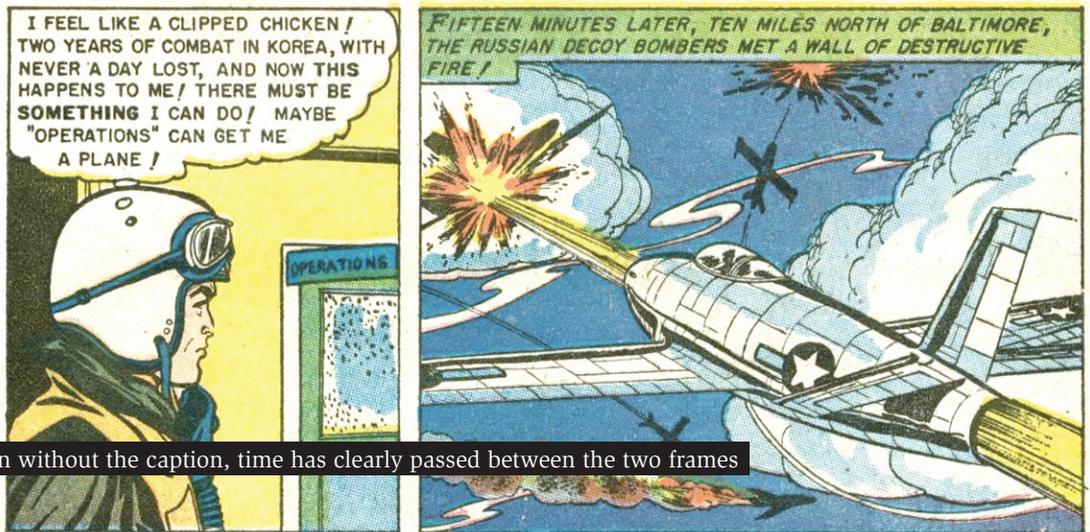
## 2.3.2.2 ¶2

The standard for understanding comics has always been Scott McCloud’s *Understanding Comics*. I guess it’s for non-artists/non-designers, cause I never liked it but everyone else does, so I still suggest it.

2.3.2.2 **Drawing time:**

2.3.2.2 ¶1 1Depicting time is an even more interesting problem. 2Again, you might think that video or prototypes are the only way to solve it, but the same problems occur; they only show a single slice at a time, and show instead of specify. 3Especially because people are





2.2.2.2 ¶2 Even without the caption, time has clearly passed between the two frames

2.3.2.2 ¶2

The comic example above is from the first issue of *Atomic War* (1952), from the now defunct Ace Comics. It was scanned in and shared on the internet at [www.goldenagecomics.co.uk](http://www.goldenagecomics.co.uk).

2.3.2.2 ¶2

poor at accurately perceiving time, for anyone to understand time in order to reproduce it, they would need to get a stopwatch.

Comic books and storyboards take advantage of innate and culturally-learned senses of reading and understanding time and space:

- 1Panels proceed, or read, in conventional language direction — left to right, but Japanese comics read right to left.
- 2The narrative flows at a reader’s individual pace; time is often non-specific, and is depicted differently when this is important.
- 3Elements that intrude or pass from one frame to the other imply more immediate time relationships; sometimes, they are actions occurring at the same time from different points of view.
- 4Breaks, gaps, time-centric annotation (“The next day”) in breaks, and establishing scenes, can indicated that additional time has passed.
- 5Panels can be of arbitrary size; this is used to indicate importance (along with shape, color, contrast, etc.) and does not imply anything about time.
- 6Panels can be ordered to allow switching between narrative threads happening concurrently; they may be switched back and forth, and keyed by color, shape or other principles, or contain insertion symbols or shapes to indicate their break from the pri-

2.3.2.2 ¶1

The “comics are about time and space” frame is printed, with permission, from Yves Bigerel’s treatise on webcomics being stuck in the paper era here: [balak01.deviantart.com/art/about-DIGITAL-COMICS-111966969](http://balak01.deviantart.com/art/about-DIGITAL-COMICS-111966969)

## 2.3.2.2 ¶2

Storyboards are excellent things, and even if you don't use them, you should know how they work. The video referenced here is by far the best description of how they work I have ever seen: [www.presentationzen.com/presentationzen/2009/01/lessons-from-the-art-of-storyboarding.html](http://www.presentationzen.com/presentationzen/2009/01/lessons-from-the-art-of-storyboarding.html). It also has some nice discussion of how to use a storyboard to sell your concept. While describing them for filmed media, the concepts are identical for interactive, and especially for interactive items that are strictly time-based or time-constrained.

## 2.3.2.2 ¶4

David Malouf has an excellent discussion titled the *Foundations of Interaction Design at Boxes & Arrows*. There is a large section (for a blog post) on time as it relates to systems, interaction and people. You can read it here: [www.bboxesandarrows.com/view/foundations-of](http://www.bboxesandarrows.com/view/foundations-of)

mary narrative.

- 7 Movement and other actions within a panel are disconnected from the passage of time indicated between panels.
- 8 Many of the same principles apply within panels, such as the position of dialogue or narrative elements, to indicate their position in relative time.

2.3.2.2 ¶3 1 Musical notation, mentioned above, also flows in a reader-oriented manner (left to right, line by line) describing an ordered reality by the placement of marks on an ordered framework of time.

2.3.2.2 ¶4 1 Similar methods can be directly applied to wireframes, flow charts, IA diagrams and other interaction design deliverables.

## 2.3.3 A Unified Modeling Language

2.3.3 ¶1 1 Any number of diagramming systems could be used as the basis for depicting interactive systems like this; flow charting, for example, can be a useful method for some diagrams.

2.3.3 ¶2 1 While the box model is about information design, this is about the interaction between informational blocks, screens or elements. 1 In that sense, it is basically just a flow chart. 1 Always remember that your job is to define, and design, the user's process flow.

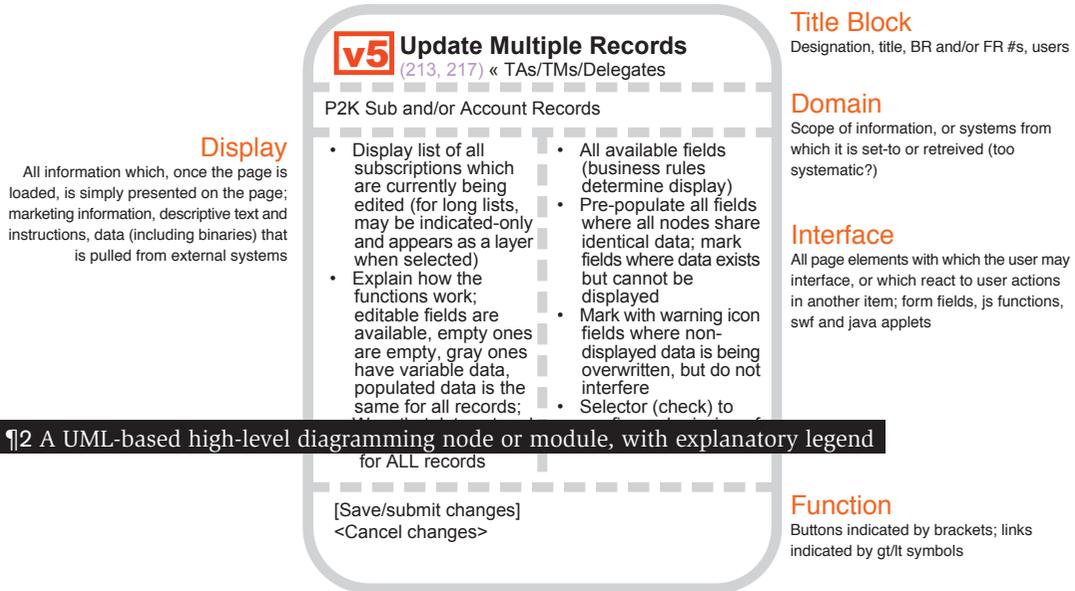
2.3.3 ¶3 1 Only discuss system points as required for your own understanding. 1 Many others will be modeling and defining systems, so you need to focus on the user. 1 Interactive design doesn't refer to an attribute of the system alone, but of the interaction between the user and the system.

### 2.3.3.1 Interaction Diagramming

2.3.3.1 ¶1 1 These modeling techniques will often be done independently, due to the complexity, and the usual timelines of most projects. 1 The whiteboard technique, especially, does work very well with a team. 1 However, unlike the box model, such exercises can take days. 1 It all depends on the product complexity, but I have spent a week, and used a whiteboard 10x40 feet to outline a process. 1 It was hard to get much of the team

to show up on Friday, even with comfortable couches and snacks. **1**Keep the scope of effort in mind when planning collaborative design sessions and accept you may do much of this alone at a computer.

**2.3.3.1 ¶2** **1**The biggest difference between this method, and a simple labeled-box flow chart is that each box has a list of elements and functions within it:



**2.3.3.1 ¶2** A UML-based high-level diagramming node or module, with explanatory legend for ALL records

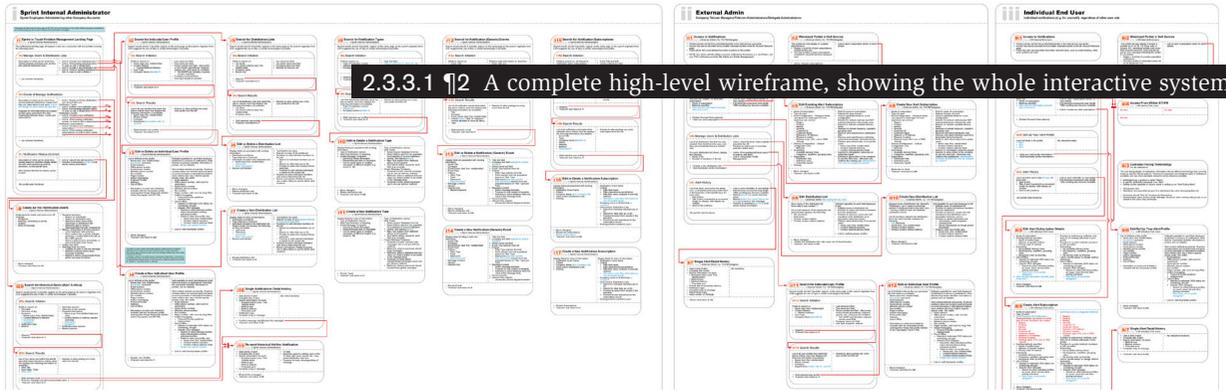
**2.3.3.1 ¶3** **1**These are based on UML (the Unified Modeling Language), specifically the statechart diagram. **2**Note that these are designed to be technology agnostic. **3**Though shown in computerized format, they can easily be drawn in this manner on a whiteboard, on paper, or even on little cards (as in the Post-It method above) for portability during the modeling process.

**2.3.2.2 ¶3**  
A simple description of musical notation, including how measures relate to actual time (albeit briefly) with nice diagrams and no need to understand music itself, is shown at: [numbera.com/musictheory/theory/notation.aspx](http://numbera.com/musictheory/theory/notation.aspx)

**2.3.3.1 ¶4** **1**The example above is a “page” in traditional design parlance. **2**However, it is much more likely to be a state of a system, a module or widget on a screen or any other interactive element. **3**See the above discussions on state and time.

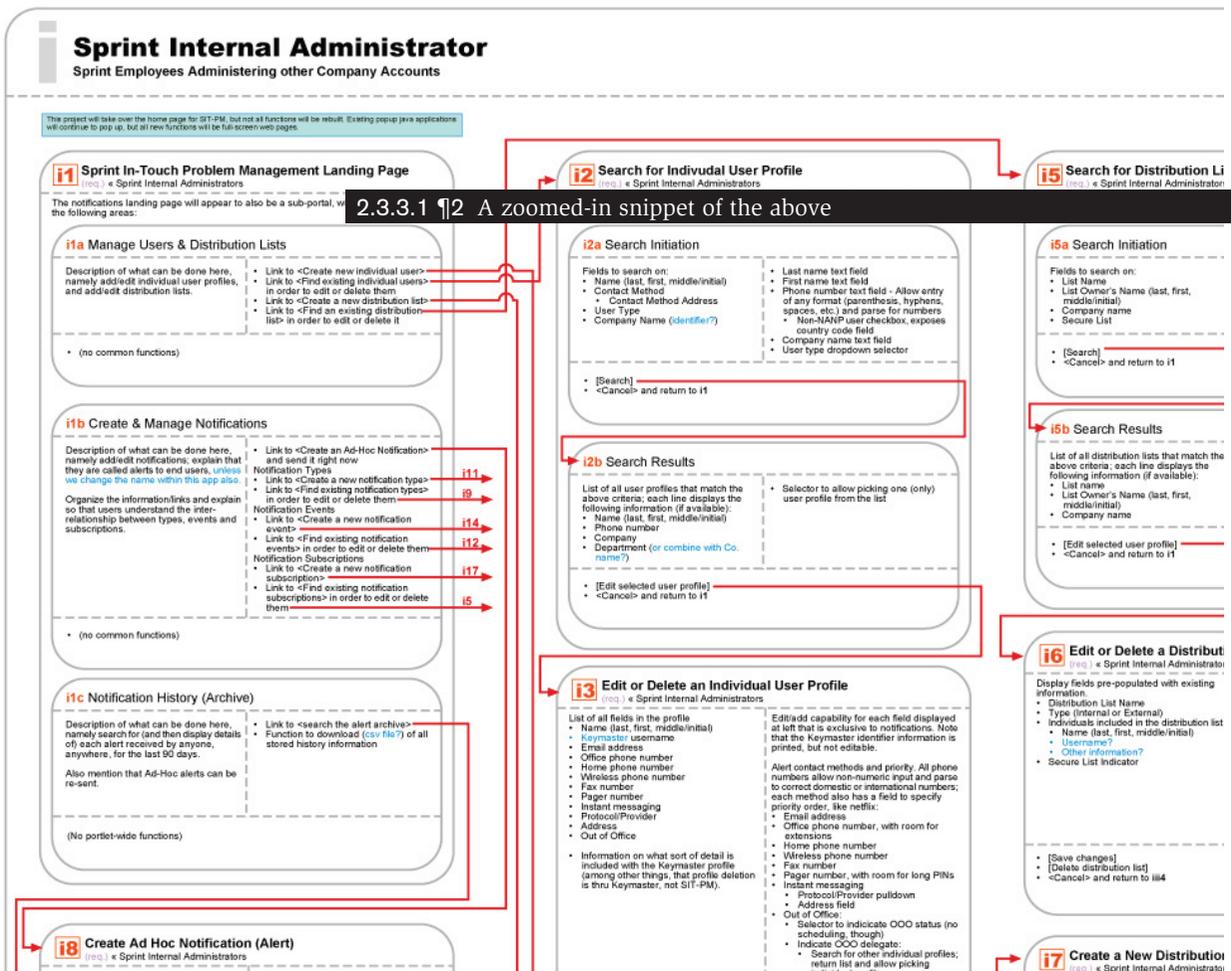
**2.3.3.1 ¶5** **1**The box — really a “node” in the system diagram — is divided into sections that define various attributes of the state. **2**The diagram above

D1319 SBS Notifications  
HLWF For SBS MetaProject Part1, (W+) - <Version 072705> July 2005 - Author: Steven Hooper



2.3.3.1 ¶2 A complete high-level wireframe, showing the whole interactive system

**D1319 SBS Notifications**  
**HLWF For SBS MetaProject Part1, (W+) - <Version 072705> July 2005 - Author: Steven Hooper**



2.3.3.1 ¶2 A zoomed-in snippet of the above

explains them pretty clearly, so I will not go into it again in the text, but the reason why is useful.

- 2.3.3.1 ¶16** 1This falls back to information design. 2Less the theory than a practical application of it for interactive systems. 3The users and domain are setting certain preconditions. 4The left side is information displayed, and the right side (and sometimes the bottom) are functions. 5Considering these as separate items allows you to view the presented or interactive information with more clarity.
- 2.3.3.1 ¶17** 1This exact method of dividing up the node is not fixed in stone. 2Certain sections disappear, and others appear. 3The node can be subdivided into sections to make optional displays or interactions clearer in some cases. 4I change them, slightly, project by project.
- 2.3.3.1 ¶18** 1In the same manner that a single element is divided into ordered sections to clearly define how it displays and functions, elements are grouped in an ordered manner. 2The UML books describe this in detail, but it tends to get a little too mathematically ordered; the theory is simple. If it's related, it's probably grouped.
- 2.3.3.1 ¶19** 1I did two basic levels of grouping. 2There can be a lot more actual nested tiers, but I mean two fundamental types. 3First, there are categories. 4The big boxes on the diagram below group basic levels of information display, and will usually be reflected in the overall information architecture, and the navigation visible to the user. 5Section ii below was not a section that users could select in this case, like a tab, but a section only exposed to certain types of users. 6Either is valid.
- 2.3.3.1 ¶10** 1They can also reflect groups of information on a single screen, such as modules on a portal-like home page (see the i1 module in the upper-left corner of the example).
- 2.3.3.1 ¶11** 1The seconds is all about state display. 2If there are multiple versions of a screen based on information available, user type or anything else, then those can be independent node-level elements, but should all be grouped in another box with its own title bar.

2.3.3.2 ¶2 s1

My favorite UML reference is “UML for Kitty Cats.” Okay, it’s an O’Reilly book, and they have chosen a housecat for the UML cover, so we always called it that (like “Ajax for Lemurs,” or whatever that is). It’s actually titled *UML in a Nutshell*, from their desktop quick reference series, written by SInan Si Alhir. While dense, each section starts out pretty comprehensible, and the diagrams are good.

2.3.3.1 ¶12 1Think about it a little and you should see how this grouping directly flows from the information design process (filter, group, prioritize...) and how interactive design related to interface design, information architecture, process and service design and so on.

2.3.3.2 UML, More or Less

2.3.3.2 ¶1 1The Unified Modeling Language is a standard notation for the modeling of real-world objects as a first step in developing an object-oriented design methodology. 2It has become an important standard language for visualizing and constructing software.

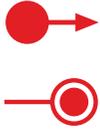
2.3.3.2 ¶2 1If you work with a software development organization that understands UML, this is ideal for their understanding, as well as looking like you are trying to speak their language. 2But in fact, it seems to be almost inherently understandable, even by laypeople.

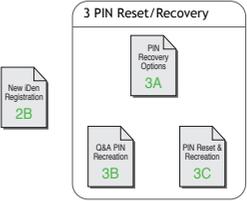
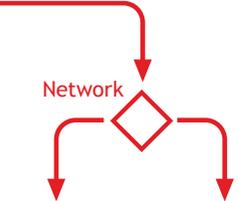
2.3.3.2 ¶3 1This is actually not all “UML” but one of several diagram styles contained within an overall framework. 2Most are much more detached from the interface, so are difficult or impossible to use for interaction design. 3However, if your product has an unusual interface (sound, haptic, control of machinery, etc.) even for a portion of the product, and it is giving you trouble, consider some of the other UML methods.

2.3.3.2 ¶4 1The following are the key diagramming elements used in my modeling method.

2.3.3.2 ¶2 s2

Other discussions of UML that are helpful can be found at the following places on the internet [www.omg.org/news/pr97/umlprimer.html](http://www.omg.org/news/pr97/umlprimer.html), [www.agilemodeling.com/artifacts/uiFlow-Diagram.htm](http://www.agilemodeling.com/artifacts/uiFlow-Diagram.htm), and [www.agilemodeling.com/artifacts/uiPrototype.htm](http://www.agilemodeling.com/artifacts/uiPrototype.htm)

<p><b>Start &amp; End</b> The beginning and end of a process is marked with these symbols. In wireframing, the end symbol denotes the end of the user’s interaction; it may not however indicate the end of system interactions. These are sometimes left out if the diagram is not overly process-oriented.</p>	
<p><b>Elements</b> Individual elements in UML may be of any required detail level, down to a single control. The box is divided into sections that label, describe printed (visible) items and indicate functional (link or submit) items.  In wireframing for web design, these usually denote a page, but can contain other types of information (e.g. emails), or other conceptual levels of information or function.</p>	

<p><b>Annotation</b></p> <p>Especially at the early stages of design, issues and questions will arise. These should be noted on the diagram itself. Annotations have varied over time and with projects, but all seem to have worked. Ideally, pink is used for issues, green or blue for general comments, etc. Sometimes this fails to work; at Sprint the brand was red for so long that pink boxes did not stick out as being bad, so I used another color instead.</p> <p>Annotations can be a box describing the information on the wireframe, a note number referring to an external text file, or even text in the design itself. Descriptions of behavior can also be annotated in this manner. Blue lines, arrows and boxes can be used to outline or point out specific areas of interest.</p>	<div data-bbox="1068 168 1258 266" style="border: 1px solid green; padding: 5px; background-color: #e0f0e0;"> <p><b>Title all notes</b> Briefly state the information and point the relevant area. Avoid all superfluous commentary</p> </div>  <p><b>1</b> My Phone - The device display block is titled with a "My Phone" label, to clarify the relationship for account authorized users.</p>
<p><b>Groupings</b></p> <p>Individual elements are grouped in surrounding boxes. These boxes are similar in layout to element boxes.</p> <p>In traditional UML, these contain grouped process items, but wireframes frequently group by logical information flow instead, or in addition to process grouping.</p> <p>Grouping can be very complex, so I suggest you read up on UML if you find it failing you. A good, practical volume is in the references in the back.</p>	
<p><b>Flow Indicators</b></p> <p>Arrows are used to describe user-perceived flow from actions performed in an element. The arrows go specific places to indicate page refreshes, new pages or new sections.</p> <p>Special symbols are used to describe joins (several arrows merging to a single point) and decisions.</p> <p>Decisions (either systematic splits or user decisions on the page before) are of the typical diamond shape with appropriate labels.</p> <p>Flow indicators should be contextual. If a page refreshes after a link is submitted, the arrows should remain in the element boundary. Element boundaries should only be broken when another page or system is referenced.</p> <p>In some cases, links will go to elements not drawn (existing items) or to a point where lines would simply add clutter. In this case, the arrow will simply leave then element box and stop at a notation of its target.</p>	

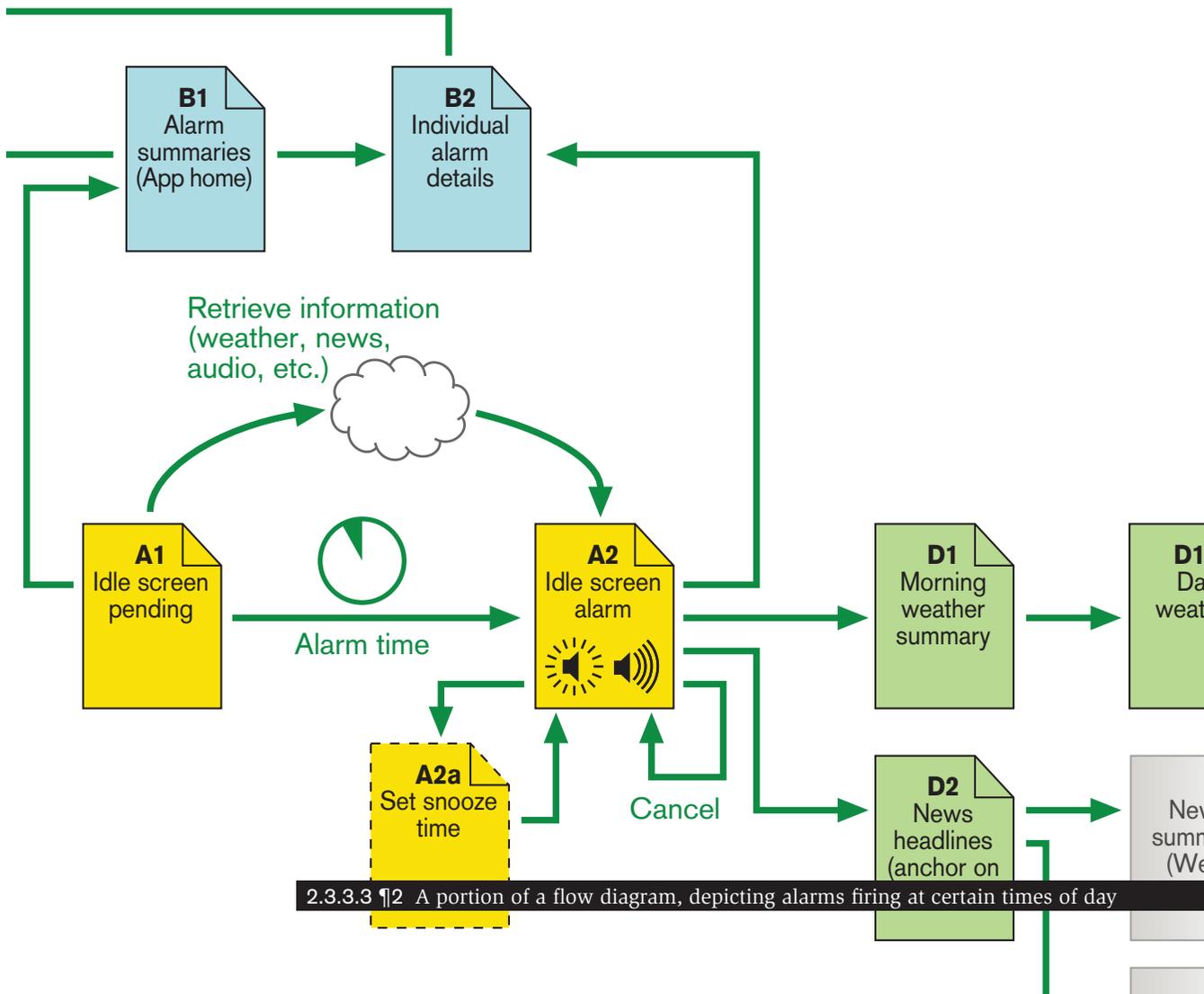
2.3.3.2 ¶5 1The above examples are indeed simply examples. 2Though the form of the elemets are often key to understanding and portability, many details of annotation, color and so on can be changed to suit your individual style, and even the style or brand of your organization.

2.3.3.2 ¶6 1Although UML is a notation system enabling people to communicate about a model, it is developed from methodologies that also describe

the processes in developing and using the model. 2As described in more detail below, the model assists with design, instead of simply being a way to record design-related activities.

2.3.3.3 **Explicit Depictions of Time**

2.3.3.3 ¶1 1There are no clear standards for regular, ordered time notation in UML or similar visual languages. 2However it is trivial to add such a notation to any diagrams. 3Most typically, delays or transition times are noted with symbols and descriptions per event.



2.3.3.3 ¶2 A portion of a flow diagram, depicting alarms firing at certain times of day

**2.3.3.3 ¶2** 1In the example, the diagram flows generally from left to right, but expressly does so when a time-bound series of actions occur. 2The clock symbol is universally understood. 3A slice to the left of 12:00 is a countdown or time-of-day, a slice to the right is a delay (other depictions are also available). 4This diagram also depicts that sound and alert tones are used in addition to visual display (indicated by the “page” icon in this high level diagram).

**2.3.3.3 ¶3** 1An implied grid of left-to-right time passage is apparent for the items visible; the retrieval of information from the cloud occurs in the background, but at the time required, instead of earlier or later. 2It is also possible to make these time ticks explicit, and create diagrams with vertical rules, labeled for specific time intervals.

## **2.3.4 Wireframes, Grids & Templates**

**2.3.4 ¶1** 1More detailed design methods, for interaction designers, tend to begin to reflect the interface in more detail. 2This is a trouble spot in several areas. 3It can step on the toes of other types of designers, but most of all it draws attention away from the interaction, the process and the user’s part in the design.

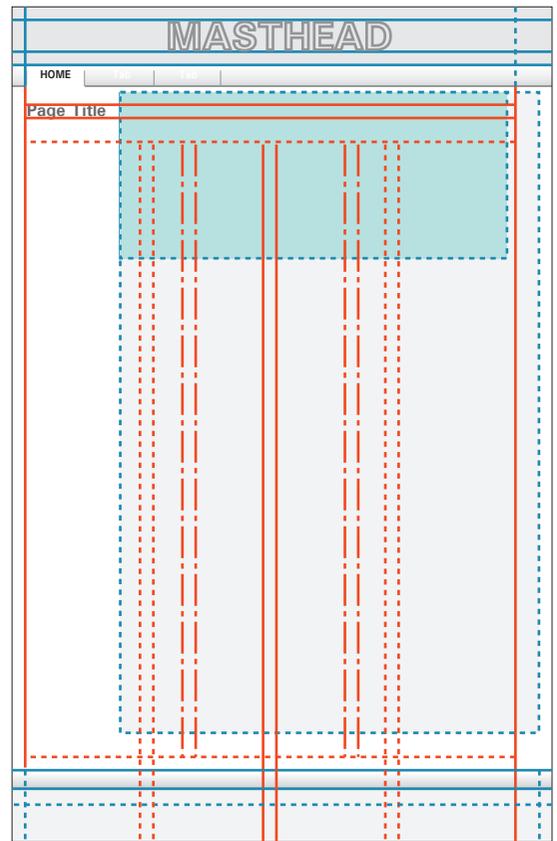
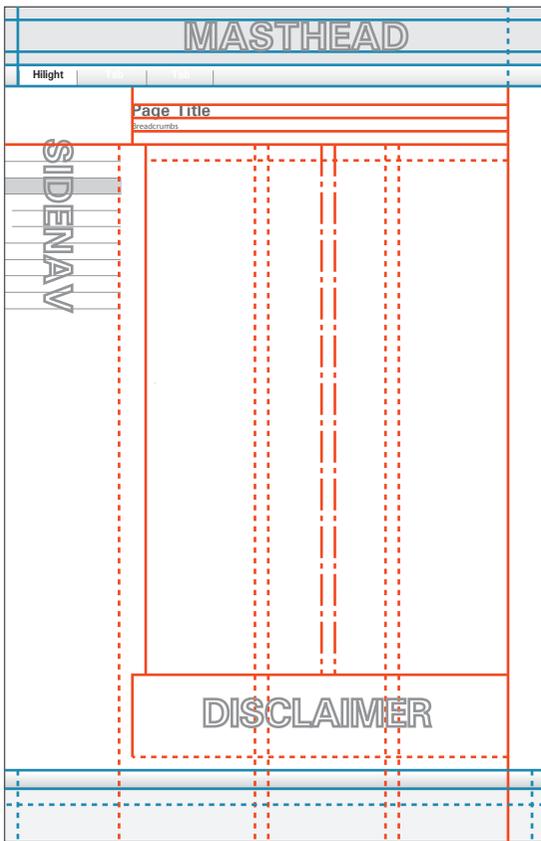
**2.3.4 ¶2** 1Collaboration and good team efforts can help to make sure that the correct teams are involved and happy with their roles. 2To avoid excessive focus on the screen design, correct modeling methods and procedures must be used.

### **2.3.4.1 Grids**

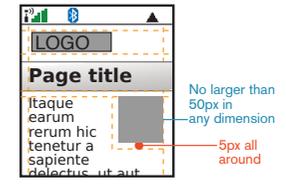
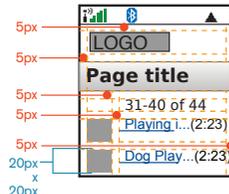
**2.3.4.1 ¶1** 1The underlying structure of each page on the site should reflect the brand. 2This includes the type, color, and critically the size and spacing of elements. 3The underlying structure of the spacing and alignment is the grid.

**2.3.4.1 ¶1**  
This is pretty much all you need to know about grids: [www.thegridsystem.org/](http://www.thegridsystem.org/) in that it’s a regularly updated archive and discussion of the whole design subset.

**2.3.4.1 ¶2** 1Interaction design elements will exist in the interface, and therefore within the grid, but the grid is not of interaction per se. 2They can flow from the information design exercise very neatly, but it’s best to get a graphic (or visual) designer to help with this step.



2.3.4.1 Two key grids used for a large consumer website



2.3.4.1 Grids for two screen sizes used for a mobile image viewing website

## 2.3.4.2.1 ¶1

Here's a decent tutorial on building armatures for making maquettes: [www.primaleyecreations.com/wire\\_armature.html](http://www.primaleyecreations.com/wire_armature.html) you don't, apparently, really use this technique for full-sized sculptures like sit in the park. I think I love that even more; even the "final" product is just a refined sketch for making a real product (out of metal). Sound familiar?

## 2.3.4.1 ¶3

1Grids are not templates and while they will be reflected in detailed wireframes, are not designed at the wireframe level.

## 2.3.4.2 Wireframes

## 2.3.4.2 ¶1

1The most used artifact you will create is the wireframe. 2Since you will be spending a lot of time drawing, explaining and revising them, this is explained in some detail.

## 2.3.4.2.1 History &amp; Background

## 2.3.4.2.1 ¶1

1The concept of the wireframe has its roots in the wire armatures underlying sculptural items constructed of clay, plaster and the like. 2A wire framework is first created, which both provides the underlying support and is often a sort of sketch of the final expected product. 3The finished form is built up and takes it's own shape, but the form always follows the wire framework.

## 2.3.4.2.1 ¶1 s2 A clay maquette dog facing its underlying armature



## 2.3.4.2.1 ¶1 s2

I cannot get anyone to return my calls or emails regarding getting an image like this, so I am just gonna credit it and pray. This is from the movie *Gerald's Last Day* by the guys who have this blog <http://justinrasch.blogspot.com/>.

If you know anyone who sculpts, please help me get a legal image for this.

## 2.3.4.2.1 ¶2

1Some web designers I know have even referred to their wireframes as a "skeletal" model, which has clear relations to the above physical sample. 2This is why I love semantics and etymology in pursuit of understanding any other topic; a wireframe is not just a simpler version, but the underlying structure of your design.

## 2.3.4.2.1 ¶3

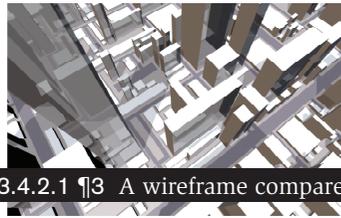
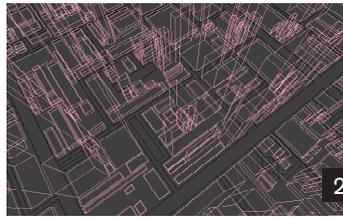
1In more recent times, 3D modeling programs have made the word more common, and in the run-together form used here. 2In this case, the wireframe is the most basic, fundamental level of viewing the modeled state of an object or scene for preview or manipulation.

## 2.3.4.2.1 ¶4

1Although various forms of representative

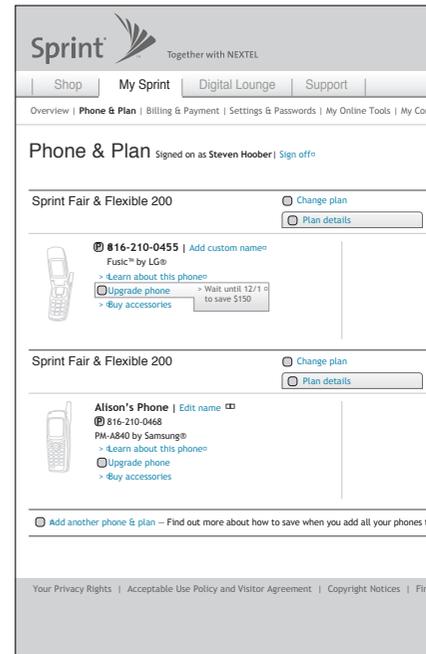
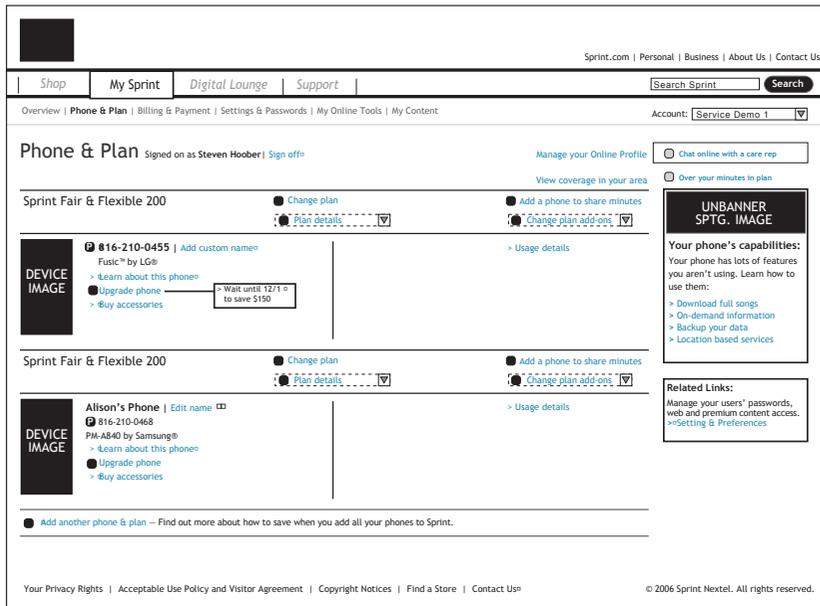
2.3.4.2 ¶4 s2

I have no references for the beginnings of wireframes for design of interactive. The exact same thing has been used absolutely forever in graphic design. Then, we called it sketching. It was no big deal. But if anyone knows who first called it “wireframing” in the web or some other interactive world, I’d love to know.



2.3.4.2.1 ¶3 A wireframe compared to a rendered of 3D scene

diagramming have existed nearly as long as has design itself, the concept of a wireframe became prevalent in UI/UE groups only since perhaps 2000 or so. <sup>2</sup>I am not sure who first used them, or where the term came to be applied, so sadly there’s some speculation. <sup>3</sup>But in my opinion, this is because of a new attention to design as a templated solution with the ascendancy of web-based hypermedia, and a need to come up with a word to describe these diagrams as different from single-page visual comps. <sup>4</sup>My team at Sprint, after previously having no particular process for this level of design, came to much the same conclusion. <sup>5</sup>This “detailed wireframe” model was derived to fill those specific design needs.



2.3.4.2.1 ¶2

The 3D scenes were rendered by Chester Terry, a programmer friend of mine who is slowly making an epic anime tale in his spare time.

2.3.4.2 ¶4

Wireframing can be badly misguided. Like these guys, who think drawing IA diagrams and the coding the page with crappy style and thick borders on the tables: [www.sitepoint.com/article/wire-frame-your-site/](http://www.sitepoint.com/article/wire-frame-your-site/) this is sad.

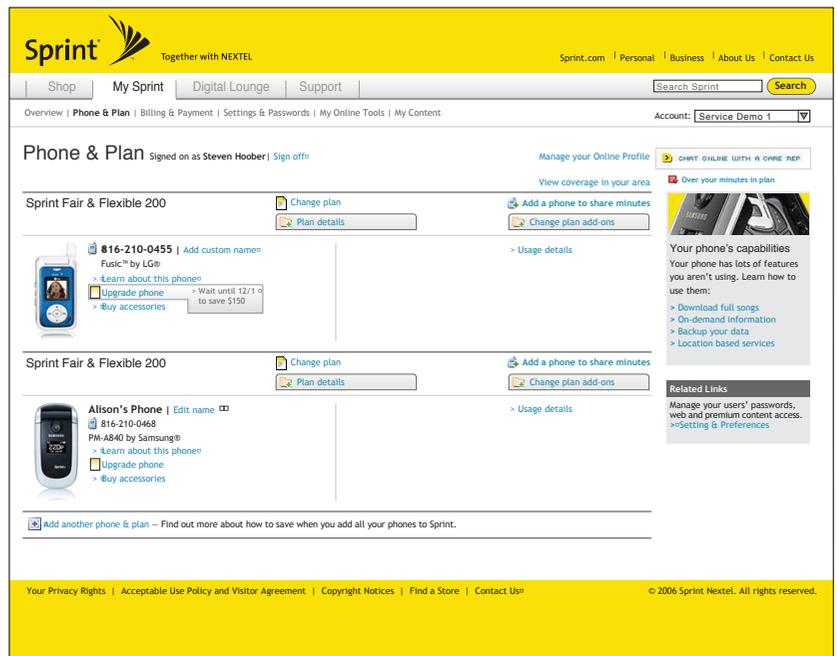
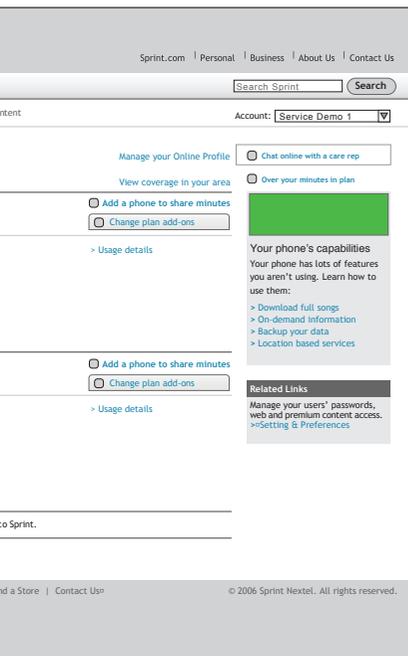
2.3.4.2.2 Enough Detail, But Not Too Much

1 Many levels of design document are called “wireframe” by various UI teams. Far too many of them, actually. 2 These range all the way from Visio flow charts to Photoshop page mockups.

1 A key to the descriptions I used above is that they all had another name. 2 What I consider to be a wireframe is not really something that otherwise exists. 3 There’s nothing else to call it.

1 Now, there are still some subtle distinctions available. 2 Most teams I have worked with (or people I have hired in) are used to building fairly vague drawings. 3 Many in fact insist that a wireframe must be a blueprint, in the sense that it has to be skeletal and one color. 4 Like the drawing on the left, in the samples below.

1 Okay, that one also has color, but just to show links more clearly. 2 I say that other levels of detail are valid, appropriate and often required. 3 A common refrain is that getting too detailed



2.3.4.2.2 ¶3-4 Three plausible levels of detail for the wireframe, all of which I find acceptable

2.3.4.2.3 ¶2-3

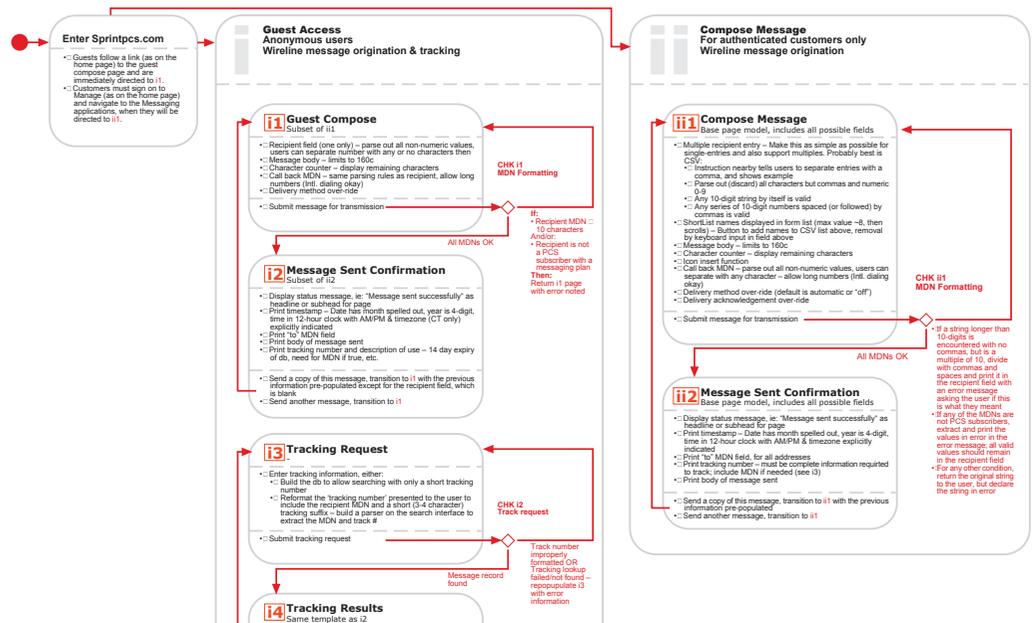
The sample showing a high level diagram and the equivalent detailed wireframe are both real, and depict a redesign I did of a (web-initiated) SMS service for Sprint in 2003. This was early in the process, and is one of those that helped me sell the whole concept of a design process to many others. The new design eliminated almost all errors, all “user-initiated errors” (there were some fifteen alertboxes the user could cause to fire just by clicking the “wrong” radio button), and even the help system. Millions of users a week, and it surpassed all metrics (completion, customer service, use, etc.) by unprecedented values. It even made me believe.

disregards visual or graphic designers, or the role of prototypers or development.

2.3.4.2.2¶5 1 Even if you have those resources (and we don’t always have them) I say, what happened to collaboration? 2 Simply drawing a rough diagram and handing it off is a waterfall process that, if you are lucky, questions that someone will come ask. 3 If you are unlucky, they will leave gaps that implementation teams will just surmise or guess at. 4 These gaps will not be what color to make things, but items that disappear from the page, or are in the wrong order in a list, or are moved from one side of the page to the other.

2.3.4.2.2¶6 1 Interaction design has to stay engaged throughout the design process. 2 That means, you have to keep improving your specification to assure that it keeps reflecting the needs of your client and your user based on your analysis throughout this process.

2.3.4.2.2¶7 1 In practice, I usually do end up going almost



2.3.4.2.2 ¶2-4

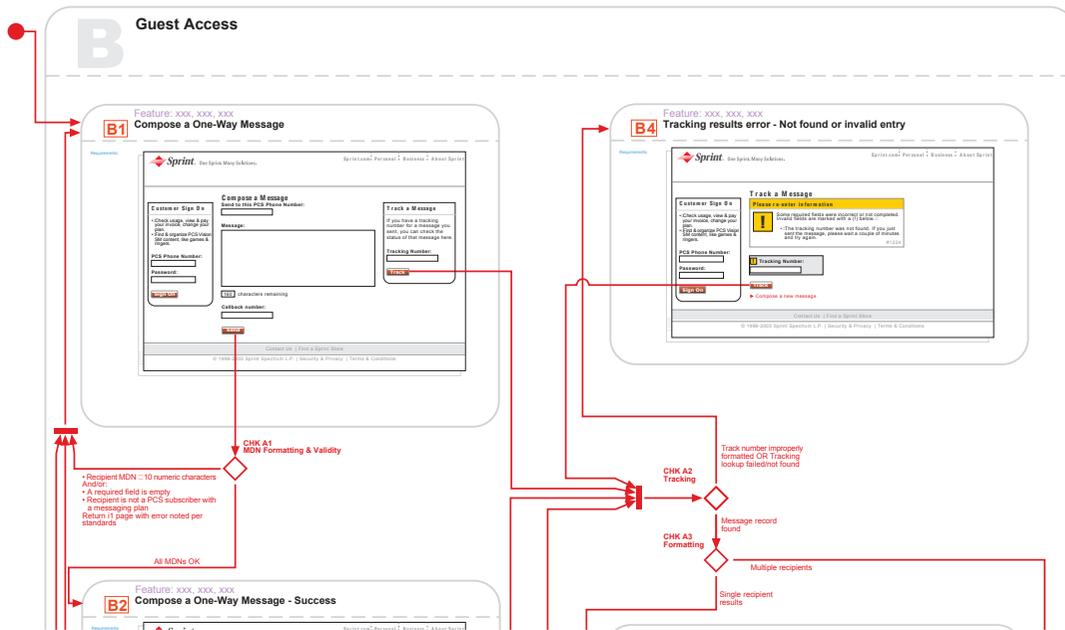
These three screens I created from some drawings I did while leading the design of the Sprint customer service portal (we didn't call it that) in mid '07. By then I was deeply into this process, so I just made the right-hand one for the actual spec, but I have (and still) designed in the more simplified manners, so just modified it to have something to directly compare the several styles.

straight to the right-most example above, and creating very detailed designs. 2The details are based on pre-determined branding and grids and other graphic standards, so are not just dreamed up by the interaction designer. 3But interaction design, and information design gets it's full due.

2.3.4.2.3 Moving From Drawing to Drawing

2.3.4.2.3¶1 1Most UI/UE teams that build wireframes model only single pages. 2Some indicate flow outside of the page with labels to other pages, (or worse yet, descriptions without the other page). 3But no others I have encountered regularly, who didn't first work with me, diagram both the layout and behavior together. 4The lack of clear behavior or interaction is a big gap in the use of these sorts of wireframes.

2.3.4.2.3¶2 1What I mean by "diagram both the layout and behavior" is that you preserve the flow between nodes (pages, states, etc.) from the UML-based



2.3.4.2.3 ¶2-3 Flow and nodes directly move from high level to detailed diagrams

high level diagramming method described above.  
 2Like the diagram on the previous page.

2.3.4.2.3¶3 1The high level diagram on the left was directly moved to a detailed diagram on the right. 2The UML-like box nodes have become wireframed screen designs, and the additional detail has added more information, sometimes more alternative methods, and often items need to be moved on the screen to fit, but it's the same drawing. 3As you moved straight from lists, to box models to computerized diagrams, each diagram flows naturally from one to the next.

2.3.4.2.3¶4 1Yes, this exact solution is sometimes incompatible with certain technical constraints on deliverables, but we'll talk about that later on.

### 2.3.4.3 Templates

2.3.4.3¶1 1Grids are often mistaken for templates, but the grid is many levels higher. 2A template can be a good thing, and I have used them, but they tend to be used incorrectly, so I am always careful in how I refer to them.

2.3.4.3¶2 1The bad way to use templates is to confuse them with grids and create them too early. 2Or worse, to borrow them from somewhere else, or from your handy 1001-web-templates book. 3Templates smack of design patterns, which can be good as long as they are used appropriately, as guidelines to fill in needs you found and defined with the rest of the design process.

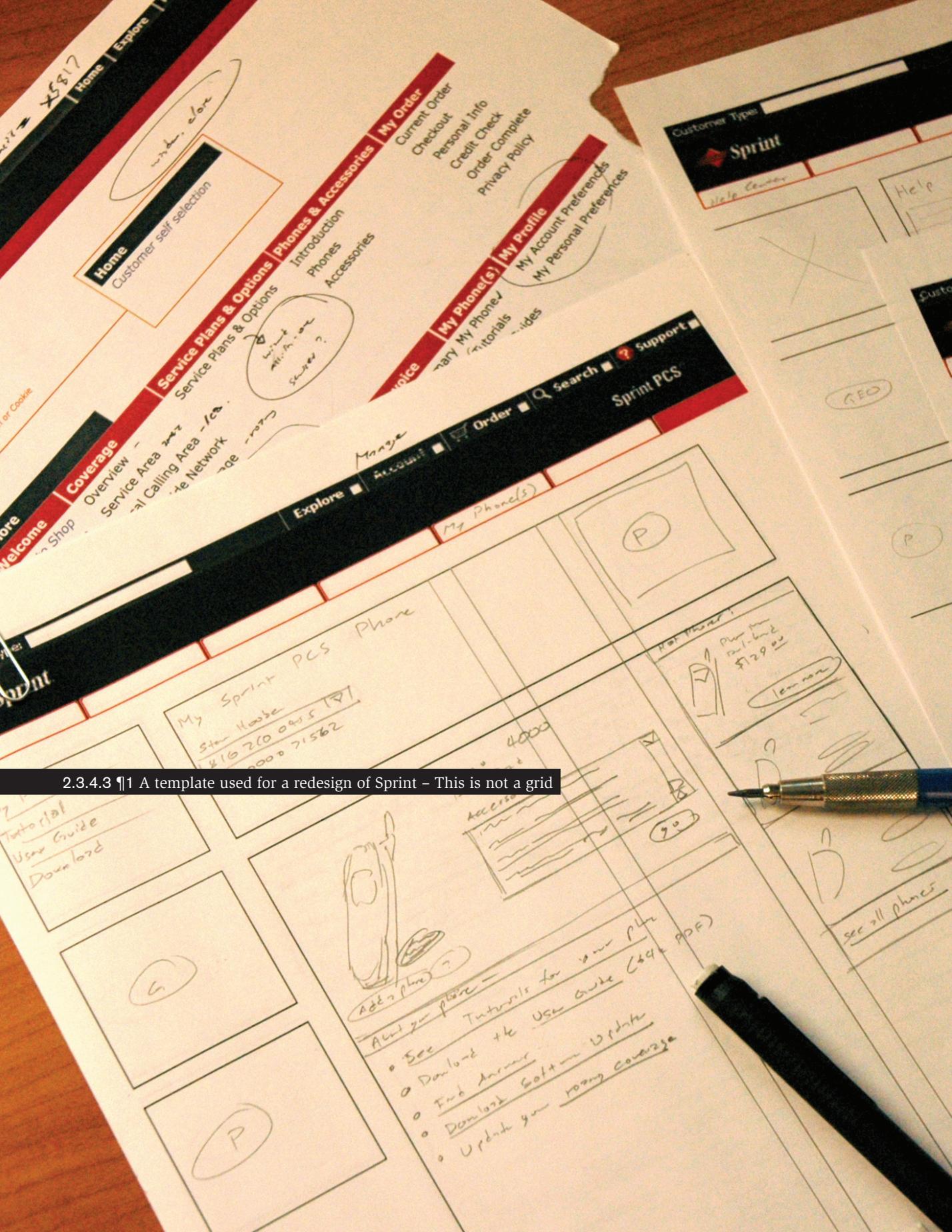
2.3.4.3¶3 1The right way to use templates, which I do sometimes encourage, is as a guideline to assist implementation teams with understanding how to save effort by re-using components on similar pages. 2A better way to do this is to directly copy development practice and design items for re-use.

### 2.3.4.4 Object Oriented Design

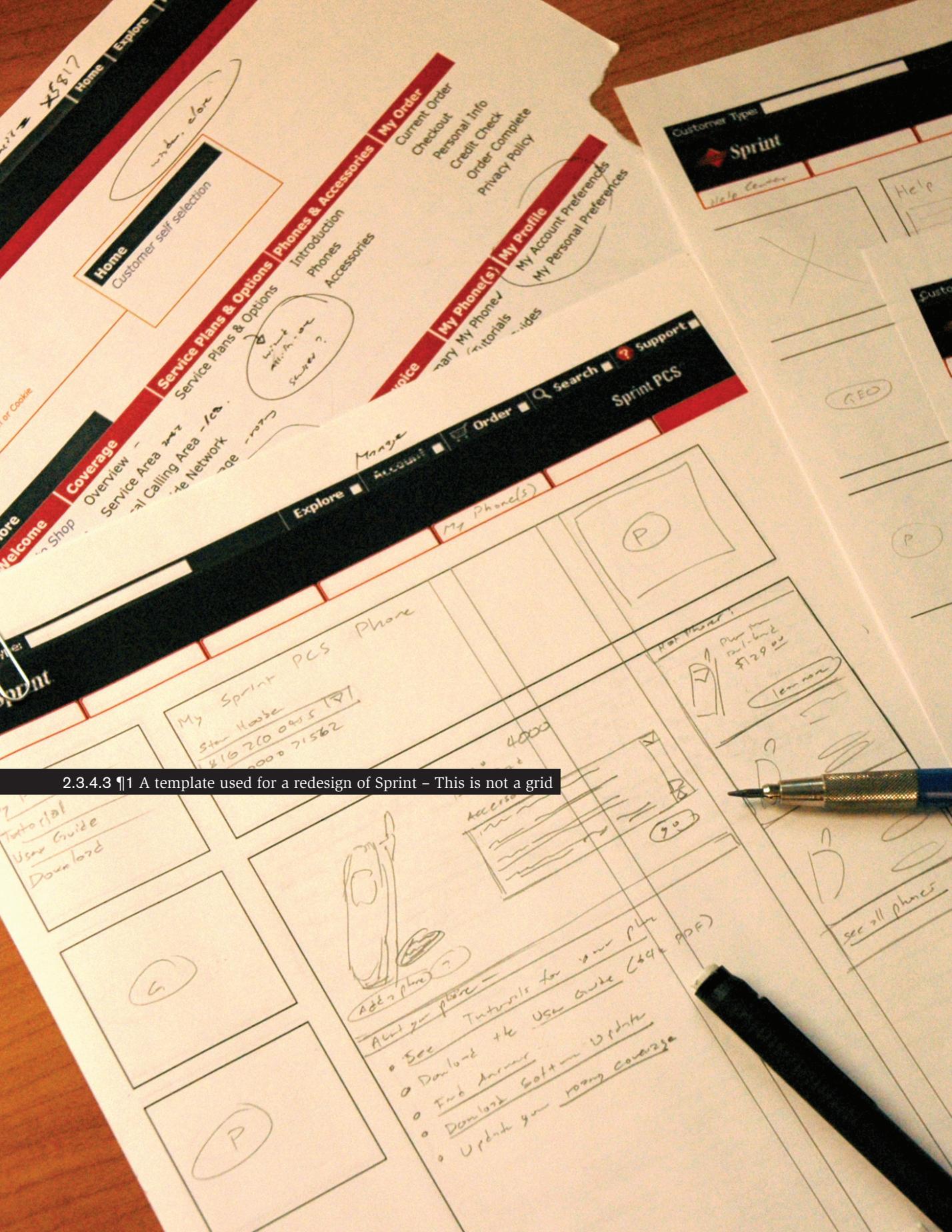
2.3.4.4¶1 1In the 1970s, the architect Christopher Alexander came up with the concept of design patterns. 2You'll note that I haven't mentioned how to attach patterns to your design in until just barely in the section above. 3That's because I don't

2.3.4.3 ¶2 s2

There is no such thing as a "1001 Web Templates" book but there are things very much like it, in print and online. Since it pains me too much to look at them, I am not going to go seek out one to reference. I have similar issues with the "great color picker tools" I see about once a month in blogs I subscribe to, which end up being pre-made palettes. If it matters, hire a graphic designer to make your templates and palettes; if it doesn't then pick any old colors or steal a template from someone else.



2.3.4.3 ¶1 A template used for a redesign of Sprint - This is not a grid



2.3.4.3 ¶1 A template used for a redesign of Sprint - This is not a grid

## 2.3.4.4 ¶1

Alexander's key works for us are *A Pattern Language*, where he laid out the keys to the development and use of design patterns, and the four book series *The Nature of Order*, in which he basically rejects the patterns concept in favor of something much more user-focused, and on processes that let you develop these. One concept thrown out is that of morphogenesis, taken from the biological discussion of the process that causes organisms to take on their shape (and presumably, a shape appropriate to their conditions and needs). A not totally unrelated conversation along those lines, also in architecture is found in the "parti" concept of a core to the design in Matthew Frederick's *101 things I learned in Architecture School*. This theory is espoused for interactive design by writer and designer Luke Wroblewski.

think a lot of patterns as typically employed.

## 2.3.4.4¶2

1 Basically, this is because of the contextual nature of everything, but especially of interactive design. 2 Take a good pattern, put it the wrong place, and it's a bad pattern. 3 Clearly, I feel that identifying needs and an analytical design process will get you there better than simple cut and paste patterns. 4 Software development has tried to apply Alexander's design pattern concept to their own work.

## 2.3.4.4¶3

1 A strict application of these patterns has some issues when compared to other componentized development philosophies. 2 But, that's getting a little detailed. Instead, I'm willing to let poor understanding work to my favor and consider all of these concepts as object-oriented programming.

## 2.3.4.4¶4

1 This is a concept that's pretty easy to understand just by its name. 2 Objects are created, they can contain other objects and they can be re-used. 3 This is an area where strictly following the concepts will help a lot in getting your product built; the developers can save lots of time and effort on the whole project if they have all these re-usable components. 4 Of course, you have to tell them they are re-usable. 5 Good drawings help this, but I guess because no one does it, no developer is used to it. 6 They will assume you have failed to provide enough screenshots, and ask for them. 7 Explain it.

## 2.3.4.4¶5

1 A key to me is that the objects (patterns in another sense) are made up for each product. 2 They are shared between projects as needed to share common information, but if you look at it the right way, any two things that re-use a presentation-layer object are part of the same product.

## 2.3.4.4¶6

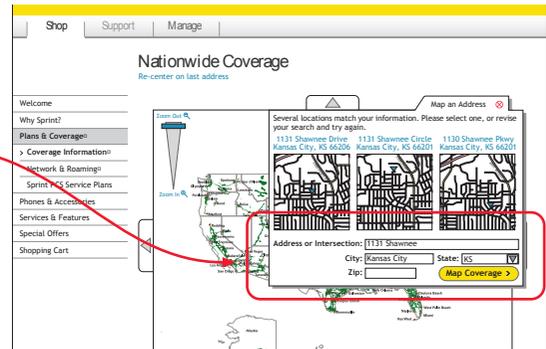
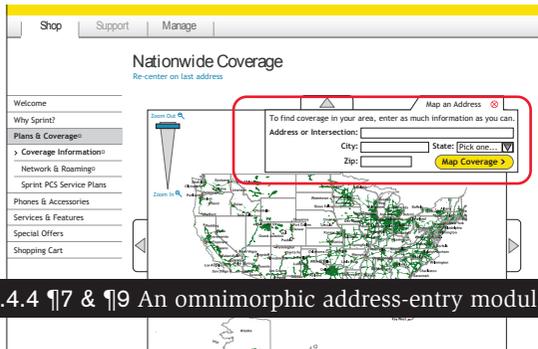
1 Re-use of an object here is critical. 2 There are two methods that are important to understand here. Remember, this is objects as designed; changes to presentation as a result of data do not apply here.

2.3.4.4 ¶2

Yes, patterns can be useful as a sort of way to codify best practices. I like the latter term a lot more (as I think it implies they are...

2.3.4.4 ¶7

**1Omnimorphism** — Universal change. **2**Any time you use an object, it's the same as all other objects. **3**If you need to change it to fit in a space, then that change has to be made to all instances. **4**Sometimes this works, and for design purposes sometimes it's great. **5**Obviously, it is also very limiting, so sometimes it does not apply well to a particular design solution.

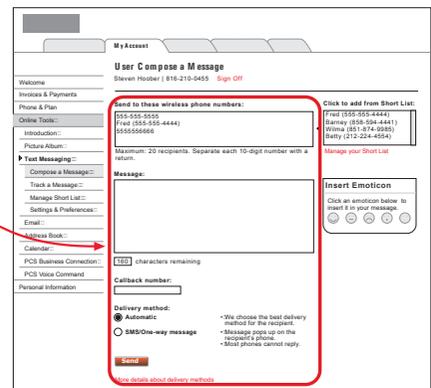
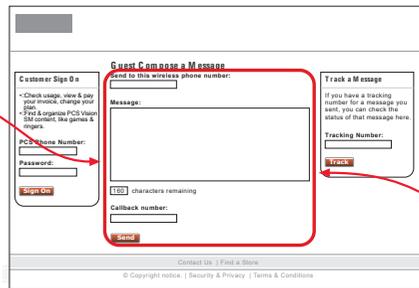


2.3.4.4 ¶7 & ¶9 An omnimorphic address-entry module

...guidelines) but it annoys some people who feel semantically the opposite about it all. I have even posted lots of patterns for mobile to the design4mobile.mobi/patterns library, but

2.3.4.4 ¶8

**1Polymorphism** — Individualized changes. **2**Any instance of an object can have a unique variation from the core design of the object. **3**If you need a special version of an object, make it up. **4**But keep in mind that there has to remain a set of core functions. **5**In the realm of interface design, its best to change sizes and shapes, add and remove services and so on. **6**As opposed to modifying the core service offered by an object. **7**This is important in design, as well as programming, so is not an issue, though.



2.3.4.4 ¶8 A polymorphic web-originated sms entry module

## 2.3.4.4.4 ¶2

I have no references for the beginnings of wireframes for design of interactive. The exact same thing has been used absolutely forever in graphic design. Then, we called it sketching. It was no big deal. But if anyone knows who first called it “wireframing” in the web or some other interactive world, I’d love to know.

2.3.4.4¶9 1For example, you have a service based on location information being entered by the customer. 2It happens several places, and in different conditions including during error states, such as when the customer may choose from optional addresses. 3However, the core information entered, or read back, is always in address format. 4That section, the address field, labels and button to submit it never changes regardless of how it is used.

2.3.4.4¶10 1Late in design, development reminds you of a logic problem with parsing free-entry addresses. 2So, you break up each address element into its own field. But this change happens for all instances of the address module. 3This is a good omnimorphic element.

2.3.4.4¶11 1Now consider instead the web-originated SMS service above. 2It was exposed to both registered users and guests, in both portlets and in a full-page view with additional features.

2.3.4.4¶12 1The message composition module is always the message composition module, but what features, functions and even what layout it has varies. 2Though these changes are sometimes data driven, based on the user profile, it’s not a data change, but a presentational, functional change. 3Each variation can be used as much as it is needed, or a switch can be made to another variant.

2.3.4.4¶13 1This is a polymorphic object.

2.3.4.4¶14 1Note that the UML (as discussed in a previous chapter) is a modeling method that well-supports object-oriented design. 2There are no clear-cut, tool-agnostic methods to universally update omnimorphic changes, but if this is a problem, it might be worth your time to explore library or template functions in your drawing tool.

## 2.3.4.2 ¶1

The photo of the templates is posed for this book, but the paper is all real. It's a strict use of a template for design purposes, and is something I did way back before I had a solid process. I made, after the work I discuss in the text with creating mockup designs and brand standards, a grid and a series of templates, which I then printed out around a hundred times. Then I got to sit in a small room while the project manager walked technical teams and product owners through the room... for up to 16 hours a day for several months. Except for the time I almost got into a fistfight with a stakeholder, this mostly impressed people with the fact that design could be so integrated into product development. It made me sad as overtly inefficient, and is one of the reasons I started looking into better processes.

# Section 3

# **Artifacts**

### 3.1 Drawings are Objects

- 3.1¶1 1Artifacts, in the sense we are discussing them, are items that are have physical presence, and are designed to be permanent or persistent.
- 3.1¶2 1This might seem like a simple reiteration of what I have been saying all along about getting your thoughts from your head to paper, but it's a little more than that. 2For example, a whiteboard is not an artifact. 3It's not permanent, and not really designed to be so.
- 3.1¶3 1A key factor in true permanence is, now, the ability to reproduce and distribute. 2So for our purposes, we are discussing documents that can be distributed to the project team members and those who need to execute the design, as well as how to actually share documents and communicate your design intent.

### 3.2 Using Artifacts to Design

- 3.2¶1 1Let's get back to process for a bit. 2Basically once you toss everything into the computerized diagrams, you should continue to re-analyze them by referring to the design objectives (and the info that fed into that, like problem statements and morphology) and following the info design process.
- 3.2¶2 1This is easier when (as in the UML-inspired diagramming method described above) you have a list of information presented which is separate from the functions. 2Simply sort through each of them in turn, and then compare functions to information and make sure it all works together.
- 3.2¶3 1As diagrams become more final and realistic, and even for prototypes and final production items, the same technique can and should be used. 2You will just have to use more brain-power to interpret what the core truth is behind each designed feature. 3Continuing to maintain earlier design artifacts can help with this.

### 3.3 Concept of Design

- 3.3¶1 1The first of these “final“ artifacts is the last one I have developed. 2The name, and it’s intent, is derived from military and operational planning, as well as from finally listening to my years of minor problems in communications.
- 3.3¶2 1Final deliverables of the styles outlined next worked great, and there were demonstrable improvements over time in speed, and accuracy of execution. 2Clearly, everyone knew what they were getting, and what they were supposed to do. 3But often it has been a problem at the beginning to explain what we are going to do, eventually.
- 3.3¶3 1My theory has always been to start with the least visual design possible — words — move to boxes, and flows and eventually design falls out of this. 2Usually rather nice and very appropriate design. 3However, there are clients, product owners, and politics. 4All too often those people don’t understand basically what they are signing off on based on words and boxes and you don’t get approved (or paid). 5They also have bosses with even less time, and they are simply not interested in listening to your wordy, or box-based descriptions.
- 3.3¶4 1Even the technical teams need something visual to get grounded upon. 2After showing anyone this basic view and explanation, all the charts and boxes make perfect sense and will often get reviewed robustly and accurately.
- 3.3¶5 1For a long time I muddled through based on working mostly with people on site; I’d show up and draw something on their whiteboard, and explain it. 2Then, when I had to work remotely more, I would make the same thing (usually drawn on the computer) and send it off as a slideshow-like presentation.
- 3.3¶6 1Sometimes I would also build documents like a tiny, high-speed high-level design — sharing some box models and flows — or even a few pages of a detailed design with annotations. 2Usually just to discuss a specific technical or presentational feature. 3Generally, these would become the high level document directly.

- 3.3.1.7 1Another issue that often comes up is explaining, in useful ways, some of the reasoning behind designs. 2One day I delivered a document that showed the transition from the way customers scanning for products in a bricks-and-mortar retail store, to a method of enabling the same scanning method for interactive product browsing.
- 3.3.1.8 1I realized I had just delivered a type of document that was different from anything else, and more importantly had value for all projects. 2It was so obvious then that I have delivered one for every project since then, to great effect.
- 3.3.1.9 1Originally it was named simply a Concept Document but I quickly realized that it was a concept for the design, not the project, product, or execution, so named it to match.

### 3.3.1 **Creating and Presenting Design Options**

- 3.3.1.1 1A key reason for a concept document is to offer design options. 2I have generally avoided working in this mode, and do not discuss it above much because I just don't think like that. 3The design naturally falls to a single direction.
- 3.3.1.2 1But a lot of clients do expect to see options from which they choose, and your sales (or management) team is likely to be promising this to the clients or product owners. 2A good analogy for explaining what a concept of design looks like is that it is the artist's rendering of several new bridge designs; illustrations are the sexy part, but they also show off representative diagrams, traffic pattern changes and notes on the concept and how it is different from the others.
- 3.3.1.3 1A great way to present multiple options is to have multiple designers, or design teams, work on the same concept. 2I know almost no one has this sort of studio organization, but stop and think and you can get there. 3Unless you are truly the only designer, there will be others on the team, and brainstorming sessions can take very little of their time. 4Bring doughnuts and grab some other designers. 5Many of the procedures above, like the information design process,

will sometimes result in small, semi-arbitrary decisions. **6**If you track all of these, you'll end up with a branched design that can be turned into a series of design options.

**3.3.1¶4** **1**Be sure to present each design option. **2**Whether you have to simply email off the document, or you get to stand up in front of the team, the presentation should be the same.

- **3**Explicitly state the core concepts. The problem (or opportunity) state, the design objectives and maybe even the business requirements.
- **4**Label each option clearly as a different one. Number or letter them.
- **5**Describe, in a few words, each option, to clearly state why it is different: "Option-A: Portal layout."
- **6**Show, as a drawing or IA diagram, each option equivalently. Take about as many pages and drawing elements to show each one.
- **7**Discuss, if possible, how each concept relates to the design objectives and other core elements.

**3.3.1¶5** **1**And, keep it short. **2**Remember that it needs to be able to be viewed as an executive overview. **3**Even if the overall presentation is a bit longer, someone should be able to flip through it, or cut out pieces, so consider that when drawing and assembling it.

**3.3.1¶6** **1**Try to get the client groups (and the technical teams) to simply select a design. **2**If you get into logo-design mode, and they want "the triangle, but with the head in the middle," you will never end up with a final design. **3**But more importantly, the process will be broken down. **4**Try to explain to the team that each concept was evolved from a design process, so it's not possible to pick pieces from each and make a whole design.

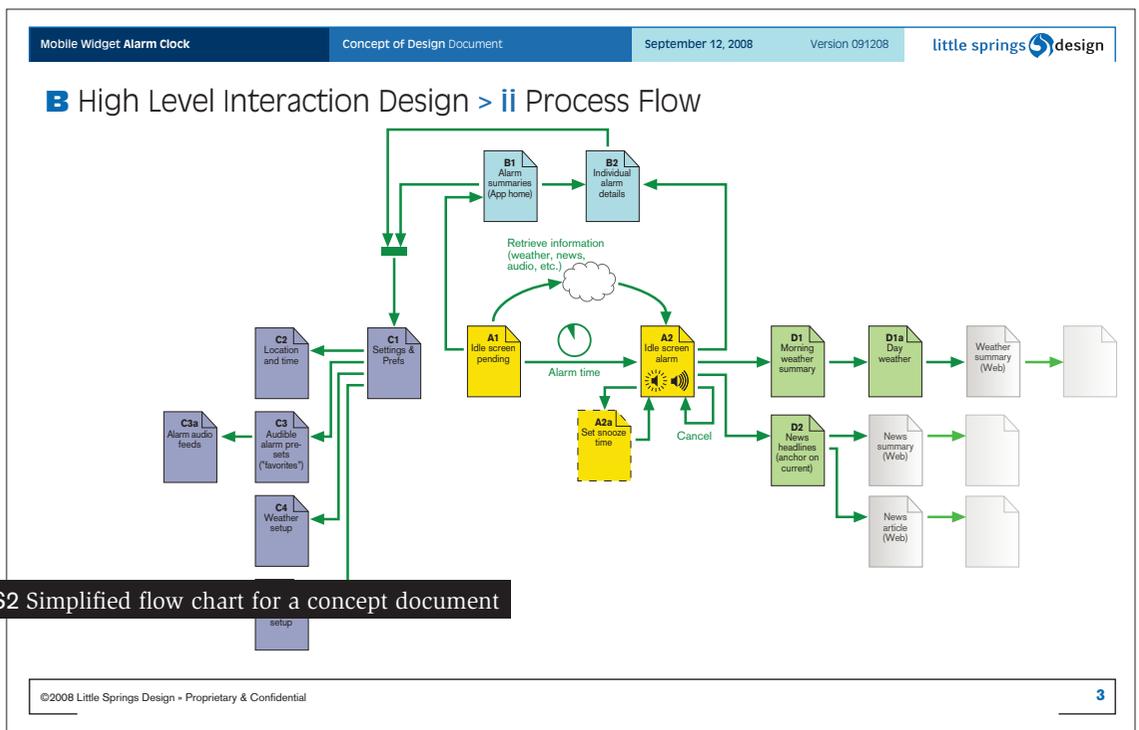
**3.3.1¶7** **1**That is not to say the graphics will be final. There should be plenty of room to play with visuals and graphic design. **2**And often, your drawing process will lead to graphics varying between different design options. **3**Emphasize and explain how interaction design is different, and easy to break.

- 3.3.1¶8 1When downselected to a single design, do not just move on. 2Complete the Concept of Design with the final concept only, and present it again. 3Make sure everyone agrees that this is the design to proceed with.

### 3.3.2 Creating Concept Documents

- 3.3.2¶1 1Although I just explained that the point of this document is to make it easier for the project team to consume than charts and words, be sure to include those charts and words. 2At least the design objectives, but any UI requirements, business requirements and even items such as the design morphology have all been successfully placed in the document.

- 3.3.2¶2 1Build it as though it is a presentation, instead of a specification. 2There can be details and notes to be zoomed in on, but make sure the titles, key phrases (like the design objectives and images can be seen and understood at a glance. 3Make sure it flows as a narrative, both on it's own and to support your presentation of the concept. 4Since you are selling this concept, or soliciting feedback on several competing con-



3.3.2 ¶4 S2 Simplified flow chart for a concept document

Mobile Widget Alarm Clock
Concept of Design Document
September 12, 2008
Version 091208
little springs design

## B High Level Interaction Design > i Information Design

### 1 Pending alarms

- Pending alarms are visible on, and selectable from the idle screen
- When available and practical, the alarm item will be visible in the "today" items,
- Otherwise, an independent widget display item will be present on the screen, and may be selected

### 2 Alarm firing

- If selected by the user at setup, an icon will appear on the screen, showing the weather for the selected morning and location.
- A text box will appear on the screen. It contains a link to the alarm details, and may contain information on any audio feed currently playing, and a news scroller, if either are selected at setup.
- For softkey devices, the cancel and snooze functions will take over the default functions
- Non-softkey devices will offer the same functions in the text box

1 Masthead

2 "Today" items

3 Pending alarm

Left Right

1 Masthead

2 "Today" items

3 Current weather

4 Text items

5 Softkeys

I'm awake Snooze

**3.3.2 ¶4 s3 Box models used to explain the concept of design**

Gray represent items extant on the device, which are unaffected by this product

©2008 Little Springs Design - Proprietary & Confidential
5

cepts, you will need to present it, even if just over the phone.

**3.3.2¶3** 1Other than that, use anything you want to communicate the concept.

**3.3.2¶4** 1I almost always include flow charts and box models of the most key screens or templates. 2The flow chart does help with an overall picture of the design, and doesn't need to be complete to serve it's purpose here. 3Box models often need explanation, but I like to include them to show that there is meaning behind designs. 4This can short circuit lots of challenges and micro-management of item placement.

**3.3.2¶5** 1The expected deliverable will be detailed page designs. 2Usually, the home page (see discussion of that in the first section). 3Since this is supposed to communicate interactive design, not graphic design or illustration, I make a point to show such drawings in a process. 4Come up with a likely use case; the notes can even personalize it quite a bit. Make sure the client can relate and stick to this case when walking through the document.

**3.3.2¶6** 1It can be useful to add other types of examples. 2A favorite of mine is the sketch style, with a line drawing of the user and environment, and a detailed drawing of the device, usually in one of the modes from the detailed design process flow to the left.

**3.3.2¶7** 1Find your favorite illustrative style — whether it is photographic, infographic, storyboard or something else — and use that to explain your idea. 2I have even used video with narration to explain concept at this level, but the variation in documentation style (not something printable on paper) makes this sort of artifact challenging. 3Prototypes can be similarly troublesome, as well as inviting all the problems I outlined in the first section with early-phase prototypes.

**3.3.2¶8** 1The document is designed for people who care about the project, so can go into some detail, and make some assumptions, but make sure there is at least one page or element that can be pulled out for executives to show off how terrific the product is at a glance. 2Try to design this panel so your mom can understand it.

Mobile Widget Alarm Clock
Concept of Design Document
September 12, 2008
Version 091208
little springs design

### Functional Design > ii Detailed Designs

**1 Pending alarms**

- Depicts the pending alarm not in the "today" area
- Branded sense is preserved, with color and shape similar to the alarm state
- When available, allow direct selection to edit alarm; if not, remove link to edit and user must find application themselves

**2 Alarm firing**

- Moments before the device emits sound (whether alarm tone or audio playback) all visible widget items will load, and then the screen will light up
- The screen is designed for maximum readability, even without glasses, in the dark; overall light output is as low as practical through use of dark backgrounds
- Individual elements are visually separated, but can be freely scrolled between; the default idle screen may be inaccessible (as shown) or these may share space

**3 Detailed information**

- Additional information should be available within the widget, though this will move to a full-screen display
- Back will return to the widget-laden idle screen
- More will load the associated detail page (e.g. Weather.com) for the selected information, in the default browser
- Text information, at least, should be pre-loaded so there is no delay in loading this information

**4 Set up alarm**

- Alarm setup will default to the most likely values, like 7am, weekdays, with a "wake up" label
- Time selection is via individual scroll and moving up and down (by 15 minute increments) or via direct typing
- Recurrence includes "just once" and will provide for a "pick days" option, which opens the days of week list in place
- Display items are selected on setup, and can only be disabled here
- Scrolling down the entire page is not required – a softkey or other mechanism will allow saving immediately upon change

**3.3.2 ¶5 A selection of detailed designs, used as a storyboard to communicate the concept**

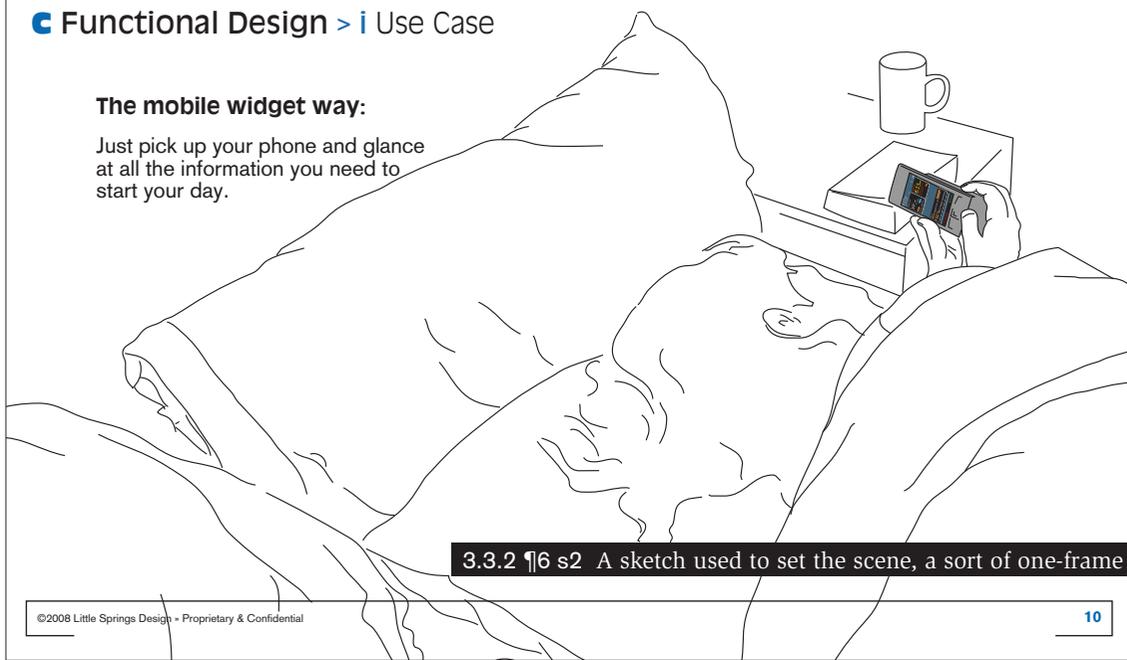
  Gray represents items extant on the device, which are unaffected by this product

©2008 Little Springs Design • Proprietary & Confidential
9

Mobile Widget Alarm Clock    Concept of Design Document    September 12, 2008    Version 091208    little springs design

### Functional Design > i Use Case

**The mobile widget way:**  
Just pick up your phone and glance at all the information you need to start your day.



3.3.2 ¶6 s2 A sketch used to set the scene, a sort of one-frame storyboard

©2008 Little Springs Design • Proprietary & Confidential    10

### 3.3.3 Synching and Revising

- 3.3.3¶1 1Following the design procedures outlined here will cause much analysis and therefore changes after the creation of the concept document. 2Since the concept document has set a sort of baseline understanding for the team, serious changes can often confuse them.
- 3.3.3¶2 1You will therefore usually want to keep this document in synch with the High Level document. 2Provide for the time to change the concept document to keep it up to date. 3Remember also that serious changes will need to be communicated, and maybe even sold again to the team.
- 3.3.3¶3 1If there is a risk to offering this concept document, it is that the project team can start getting possessive of the initial design concept, and resist changes as the design proceeds.
- 3.3.3¶4 1When the detailed design starts, you will generally abandon the CoD. 2Take parts and use them in it, but no longer update it. 3Make sure everyone knows this is true when you move to the detailed design document.

## 3.4 High Level Interaction Design

3.4¶1 1Once the overall goal is set and the project team is onboard, get back to the working documents you have already created. 2Open the box models, and turn them into a formal, deliverable document.

3.4¶2 1Once you are used to working in this manner, your box models may begin to resemble high level design documents from the start; it's not required, but it's also not bad, as long as you don't feel too constrained by the style.

### 3.4.1 From Sketching to Specifying

3.4.1¶1 1But for now, let's back up all the way to the middle of the information design process. 2Assume you have not completed it, or have only come away with a variety of sketches, Post-Its on chart paper and photos of whiteboards. 3How do you get from notes and sketches to a specification diagram?

#### 3.4.1.1 More Box Models

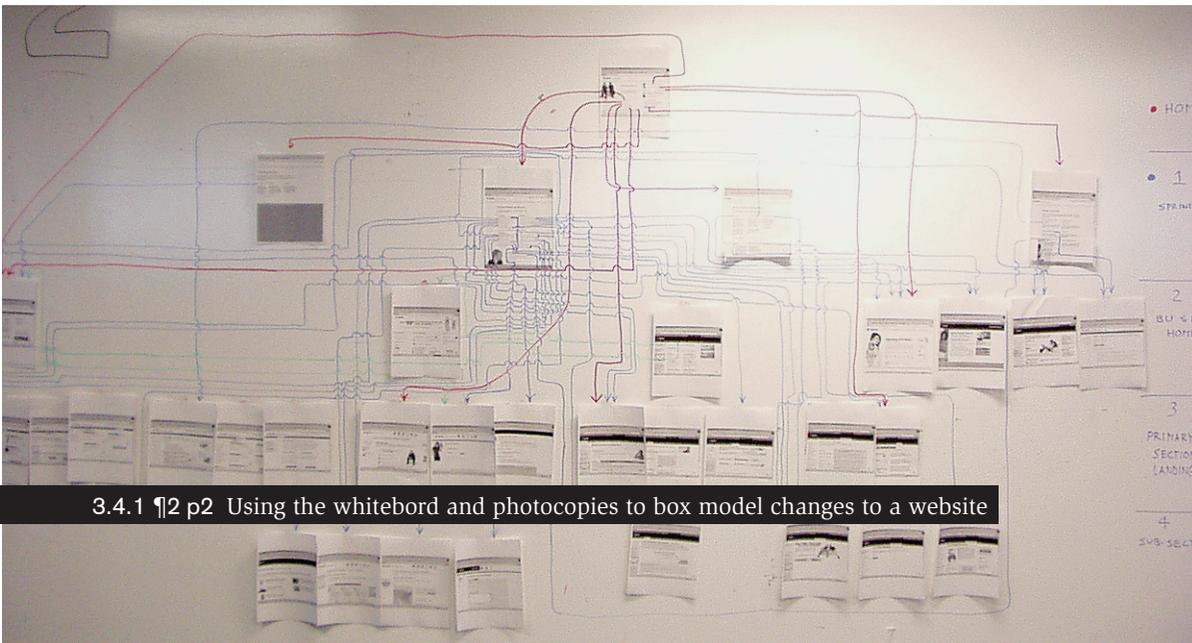
3.4.1.1¶1 1First, I'd probably keep sketching. 2I still think that the process of drawing, even of drawing detailed diagrams for specifications, is something that helps the design itself become better. 3So moving towards the final diagram in an engaging manner is something I always encourage. 4Time is probably the biggest issue, but if you can take the time, try to.

3.4.1.1¶2 1A good first run is to continue the information design process as outlined above to encompass the entire system. 2The drawing shown is a modification of an existing system, so has sketched-up current screenshots arranged in a wholly new manner. 3While not exactly the same process, it's similar and you can see the scale involved (this whiteboard is 10 ft tall).

3.4.1.1¶3 1Such a diagram can take several days to complete, especially since you will want to take breaks in between to make sure you don't go insane from it. 2It can be hard to keep everyone on the project team engaged, so instead of having everyone stay in the room for this whole

3.4.1

When I started doing high level flow charting and wireframing (and detail for that matter) there was nothing written outside of technical and systems modeling. Now there is. All the processes I talk about (even UML!) and many, many others are regularly submitted to *Wireframe Magazine*, really a blog here [wireframes.linowski.ca/](http://wireframes.linowski.ca/) If you don't love what I have, then you ought to be able to find something you like over there instead.



3.4.1 ¶2 p2 Using the whiteboard and photocopies to box model changes to a website

### 3.4.1.2 Moving to Computers

3.4.1.2¶1 1Since you will have to make this a permanent, versionable, shareable file, you need to get it into a computer drawing program at some time.

2The first step in this is remembering this is something you will need to do. 3When drawing the model on the whiteboard, or making post-it boards, remember to keep items as discrete nodes and useful groupings. 4Yes, like the UML-based modeling method.

3.4.1.2¶2 1The high level diagram will use this method, and can even employ it on paper or whiteboard, so begin to move that direction as early as possible. 2In the example, the boxes on the whiteboard all have distinct titles, and special conditions are explicitly listed.

3.4.1.2¶3 1Moving to the computer is just a case of directly drawing a node box for each element, and adding whatever notes about content you can come up with immediately. 2Speed is important here. It should be like copying to another whiteboard, like rapidly sketching the original drawing again. 3Later, you can come back in and fill in additional details.

3.4.1.2¶4 1Fight the urge to draw pictures. 2Even if you

have done so as a shorthand for previous sketching, like the phone image in the above, keep the high level diagram strict to the process. **3**The relationship between drawing styles, comparing items you have typed with those you have to draw, is key to forcing you to evaluate designs. **4**Mixing the two too early will shortcut that quality check.

### 3.4.1.3 Process, Flow and Interaction

**3.4.1.3¶1** **1**Once the boxes are transcribed, all those flow lines from the whiteboard come into play. **2**Even if you have gone light on them before this, it's time to tie everything together. **3**Use the UML-style flow system, show the important process and user decision points. **4**Don't go overboard on system processes, as they will be defined by others; just map that which directly effects the user as they proceed through the system.

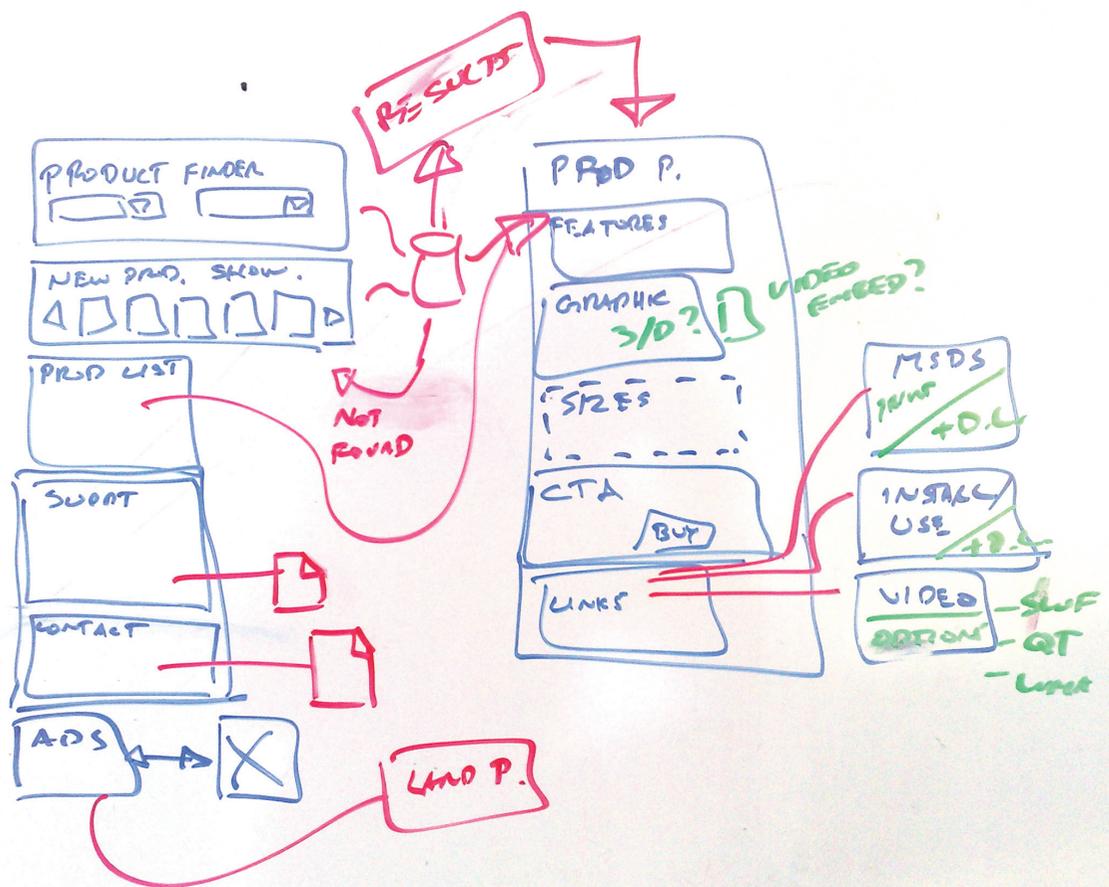
**3.4.1.3¶2** **1**This is a key step, and will probably take a little while to complete. **2**That's because you must not just draw the lines, but make sure it all makes sense.

**3.4.1.3¶3** **1**Mathematicians have told me that you can write a formula on a board (these are big formulas and proofs) and some of them can tell at a glance that it is wrong. **2**Long before you find out what exactly is wrong, there is a gestalt sense of right and wrong to the math.

**3.4.1.3¶4** **1**Similarly, while you need to trace each element to it's end, and make sure each and every use case can be performed without error or loop, the value of this drawing style, and of this portion of the process is in getting everything about the interaction onto one place so you can step back (or zoom out) and see the system as a whole. **2**If it's messy or cluttered, it might just be the way you drew. **3**But if it's "wrong" in some indefinable way, it's quite likely to be actually wrong. **4**Seek out what that is and try to fix it.

### 3.4.1.4 Details and Specifics

**3.4.1.4¶1** **1**Detail is sometimes hard for people to get right at the high level. **2**It's actually a reason I like the strict model method, as it acts as a sort of check-



### 3.4.1.2 ¶2 Moving from a whiteboard model (above) directly to a UML-based computer model (below)

## A Home

### A1 Product finder

Information:	Behaviors & functions:
<ul style="list-style-type: none"> <li>Labels for each field</li> <li>Description of the function of the module</li> <li>When no selections are made, the search button is "grayed out" and unavailable</li> </ul>	<ul style="list-style-type: none"> <li>Selector with solution categories, defaulted to "select one..."</li> <li>Selector with industry categories, defaulted to "select one..."</li> <li>[Search] for products, only becomes active when at least one selector has input</li> </ul>

### A2 New product showcase

Information:	Behaviors & functions:
<ul style="list-style-type: none"> <li>Series of items display in a scrolling list</li> <li>For each item: <ul style="list-style-type: none"> <li>Display product image</li> <li>Display product title/name</li> <li>Display link to view more details</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Selecting any individual item will load the product landing page</li> <li>Automatic scrolling TBD</li> <li>Manual scroll control TBD</li> </ul>

### A3 Product list

Information:	Behaviors & functions:
<ul style="list-style-type: none"> <li>Display name of product</li> <li>Display (icon?) for special uses, such as fire-resistance</li> <li>Clearly communicate that each item is a link</li> </ul>	<ul style="list-style-type: none"> <li>Link from each line item to the product landing page.</li> </ul>

## B Product finder

### B1 Search results

Information:	Behaviors & functions:
<ul style="list-style-type: none"> <li>Series of items display in a scrolling list</li> <li>For each item: <ul style="list-style-type: none"> <li>Display product image thumbnail</li> <li>Display product title/name</li> <li>Display category of product, both in text and graphically</li> <li>Display link to view more details</li> </ul> </li> <li>List total number of results</li> <li>Display current position in results list</li> <li>Show current search terms</li> </ul>	<ul style="list-style-type: none"> <li>Only 20 items will be shown at a time. If more than 20 results are displayed: <ul style="list-style-type: none"> <li>At the first page in a series, there is no "back" selector</li> <li>If any page other than the first in a series, "View previous 20" will reload the relevant results set</li> <li>If any page other than the last in a series, "View next n" (where n is the number of results on the last page) will reload the relevant results set</li> <li>At the last page in a series, there is no "fwd" selector</li> </ul> </li> <li>Field to enter search terms</li> <li>Button to submit new search</li> </ul>

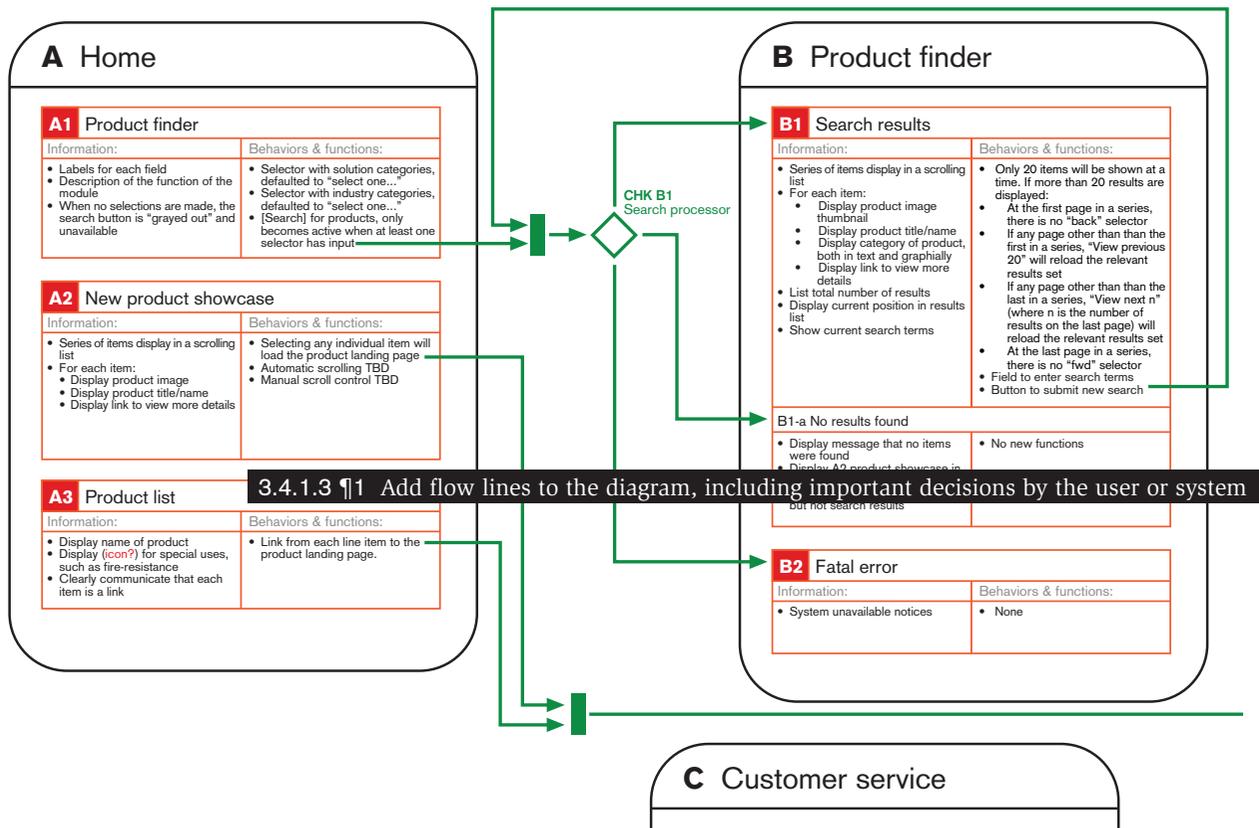
### B1-a No results found

<ul style="list-style-type: none"> <li>Display message that no items were found</li> <li>Display A2 product showcase in search results list, with label to denote they are available products, but not search results</li> </ul>	<ul style="list-style-type: none"> <li>No new functions</li> </ul>
--	--

### B2 Fatal error

Information:	Behaviors & functions:
<ul style="list-style-type: none"> <li>System unavailable notices</li> </ul>	<ul style="list-style-type: none"> <li>None</li> </ul>

## C Customer service



list to make sure everything is included.

**3.4.1.4¶2** 1The short version is: make sure everything is addressed, but not everything is included. 2Mention that all information is present, and all functions exist, where they are and how they change over time. 3But don't actually write final content or show pictures.

**3.4.1.4¶3** 1This is a specification, so be specific. 2Pretend you are writing for someone you have never met. 3Because it probably will be developed by someone you have never met, who may nor may not natively speak your language. 4So don't just say "Show name" but remember this is a specification for technical teams: "Display customer first name, a non-breaking space, and customer last name; display with the capitalization entered by the customer."

**3.4.1.4¶4** 1Much of the decision about how content is addressed for a specific product is how important it is for the service. 2Of course information is

always the key; here I mean whether the site is inherently functional and data driven, like an account management site, or more content driven like a news (or weather) site.

3.4.1.4¶5 1A content map of some sort may be appropriate to create at this level. 2Much as interaction design must not be disconnected from development of graphics, grids and brand standards, content guidelines should be tied to the high level documentation. 3This will assure there is a consistent view not just of the content creation needs, but of the technical needs for storage, indexing and retrieval of content. 4Be sure to create this content map with the input of product owners, writers and anyone else who can help. 5Content is perhaps the area where you should get the most help from those with the experience, sense of the product and institutional knowledge.

3.4.1.4¶6 1Mapping content can also help the client group outside of the current project. 2The information you gather will be like that retrieved in the col-

The Weather Channel Mobile Web 3.0
High Level Design Document
December 5, 2008
Version 120508
little springs design

## Site Content > iii

5 Health & Home		
Title	Type	Content/Crosslinks
5 Health & Home [Landing Page]	Landing	Summary of local pollen, forecast, green tip of the day; thumbnail of pet image (5.5) • Links to each subsection
5.1 School Day Forecast	Table	Same as 1.2.4
5.2 Green Tips	Text	Text, with supporting graphics as needed • Functions to access green tips archive as continuous series
5.2.1 Green Tips Archives	Text	Same as 5.2, but a previous tip; include original date
5.5 Pet Slideshow	Graphic	Series of pages, newest first, of pet images with captions and credit for submitting user • Functions to access other images

6 Photos & Fun		
Title	Type	Content/Crosslinks
6 Photos & Fun [Landing Page]	Landing	Thumbnails of photo, wallpaper; summary of almanac for the day, trivia – Prominently display Celebration weather when in season • Links to each subsection
6.1 Photos	Graphic	Series of pages, newest first, of weather images with captions and credit for submitting user • Functions to access other images
6.2 Trivia	Text	Text, with supporting graphics as needed • Functions to access trivia archive as continuous series
6.3 Almanac	Table	Historical, aggregate weather data • Function to see other dates
6.4 Wallpaper	Table	Series of pages, newest first, of weather images with captions and credit for submitting user • Link to download media to device • Functions to access other images
6.5 Celebration weather	Landing	Conditional, appears only as needed; layout changes from event to event. May include graphics, maps, text, tables and have links to other sections

3.4.1.4 ¶5 A content map in the style of a high level specification

©2008 Little Springs Design - Proprietary & Confidential

lection steps of the information design process, and might never have been explicitly laid down before. **3**By codifying and permanently defining it, the organization can follow these guidelines into the future, assuring they create content that meets the needs of the business and the user, as well as fitting into the technical constraints of the product.

**3.4.1.4** **7** **1**Content maps are generally not carried through to detailed specifications, so be sure to keep it up to date within the high level document, or to branch it out to its own document later, depending on how the client is using your design specifications.

### **3.4.1.5** **Concept Documentation**

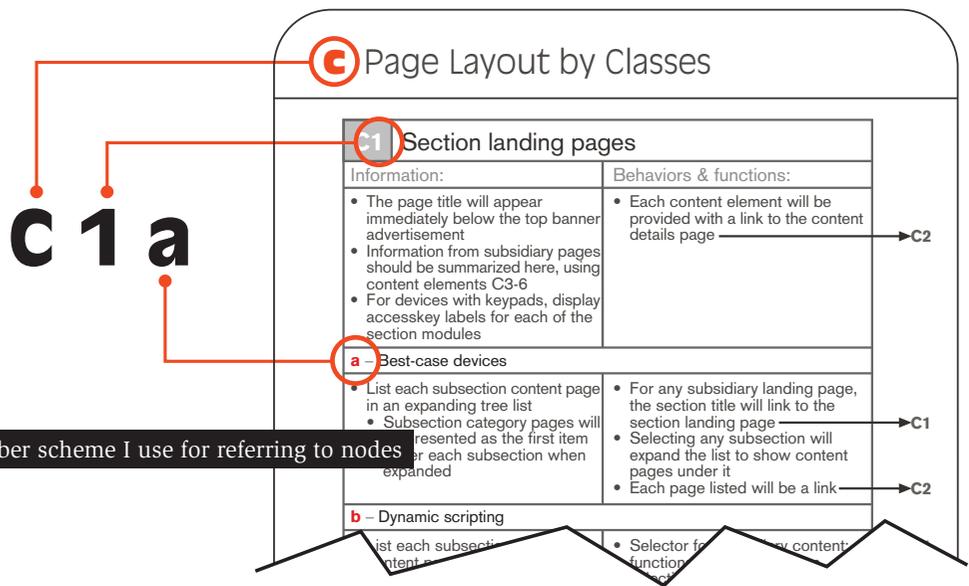
**3.4.1.5** **1** **1**It has been mentioned in passing already, but keep the concept document up to date. **2**Changes to the concept will happen as you fill in the blanks while drawing the high level design. **3**This will keep the project team (not just the clients but the implementation folks also) up to date instead of just having them think the concept is solid and unchanging.

**3.4.1.5** **2** **1**Additionally, this is a key part of that bouncing back and forth thing. **2**You write and diagram flows in the high level document, then reflect off what you wrote with visuals. **3**It serves as a check of your design, and helps you analyze your decisions.

### **3.4.2** **Numbering and Annotation**

**3.4.2** **1** **1**Each and every thing on every document you create has to be able to be referred to in a consistent way, throughout the process. **2**The best you can hope for without this is muddled requests to change the line in the left side of the second thing on page 17 of the april 21st document.

**3.4.2** **2** **1**The project lifecycle is the critical element here. **2**You have to have a consistent numbering scheme, and you need to make it so that the numbering is the same for an element throughout the process. **3**Unlike a normal outline, where items change numbers, element *A-17b* must always remain element *A-17b*.

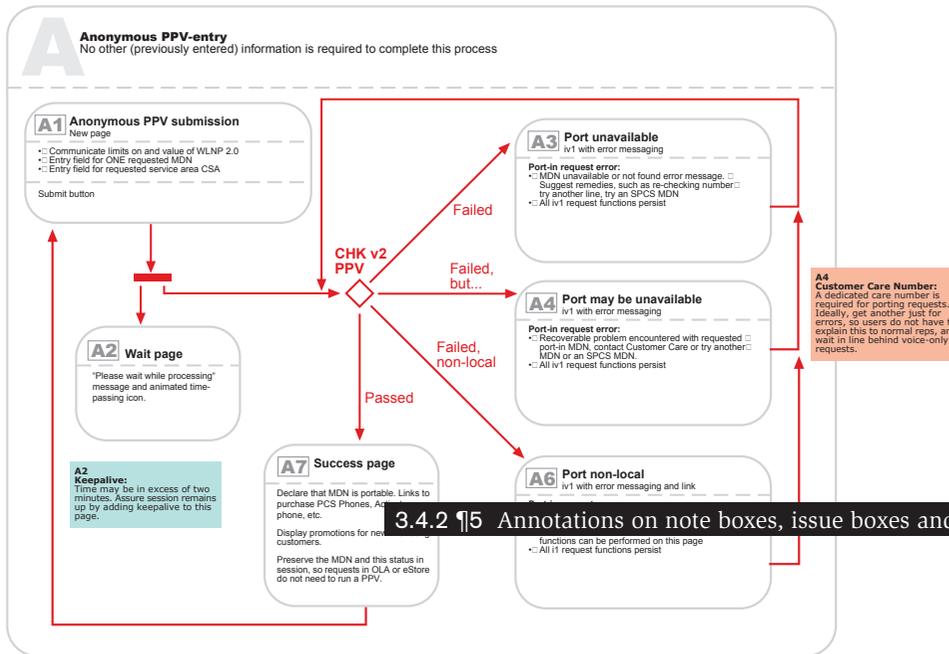


3.4.2 ¶4 The letter/number scheme I use for referring to nodes

3.4.2¶3 1This is also true between drawings. 2The high level and detailed diagrams will refer to the same elements in different ways. 3Their numbering must remain the same. 4I started doing this to work with IT better, and it sure helps a lot; being able to stand up in front of a VP and state without blinking that your team has a bullet-proof, traceable element numbering system give you some points. 5However, it's also a natural follow on to the whole concept of information vs. drawings. 6The concept of a particular element doesn't (really) ever change, so the way it is referred to should not change just because drawings have.

3.4.2¶4 1I have called this numbering, but in reality I make it more readable by switching between letters and numbers. 2Start with the largest, or highest-level grouping boxes in the UML-based diagramming method, and work down.

3.4.2¶5 1Additional annotations will be required outside of the node boxes. 2Decision points, and some other portions of the flow indicators will require annotation. 3In addition, some designs will require notes boxes as described in the UML format section. 4Ideally, there are no pink issue boxes when the document is done and delivered, but there may be blue informational boxes to communicate information above and beyond what can be in the other diagramming elements.



3.4.2 ¶5 Annotations on note boxes, issue boxes and labeling of decision points

3.4.2¶6 1The decision points also should be numbered. Keep them in a similar format as the element numbers, by section for example, but with a note of the type of element, hence the “CHK” (for check) label.

3.4.2.1 Breaking the Numbering

3.4.2.1¶1 1If you are diagramming in a way that is not yet attached to any fixed elements, is a concept or option, or is not even really a part of your project, it is usually good to specifically avoid using the numbering scheme you are using in your project, or even the format you normally use.

3.4.2.1¶2 1This is common in the Concept of Design, but is a bigger issue in the concept documents when changes are made during the later design process, when everyone starts getting used to the numbering scheme.

3.4.3 Technology of Drawing

3.4.3¶1 1The question I get asked the most about my drawings (and map making, and many other items I create) is: what program I use. It really doesn't matter — though I'll tell you in a minute

— because the concept is the most important: there is no magic solution.

3.4.3¶2 1You have to learn how to draw with the program instead. We use Macromedia Freehand. 2I cannot suggest it in good conscience now as it's been abandoned by Adobe, who now owns it. 3But we adopted it originally because it's:

- 4Multi-page
- 5Fast
- 6Simple

3.4.3¶4

If you don't understand the difference between raster (or "bitmap") and vector graphics, try to. It's important not just for drawing, but for an increasing number of internet technologies. Flash and SVG are vector, PNG and JPG are raster. Etc. If you want to read more, this is a longish set of articles on the technology differences [freerangestock.com/understanding/vector\\_bitmap/](http://freerangestock.com/understanding/vector_bitmap/).

3.4.3¶3 1Anything else that does that would be great, but since I haven't found one we are just still using Freehand. 2I'm regularly on the lookout for something that makes me happy, and brace every time there is an OS update, waiting for it to stop working.

3.4.3¶4 1Although it goes without saying that this is a vector program, it might not to everyone. 2Even though Fireworks and even portions of Photoshop have scaleable vector controls, they are designed for raster output so are not the right solution. 3There are some technical reasons around making sure the items you draw are re-usable and totally portable, but mostly this is not a graphic design creation tool, but a design modeling tool.

3.4.3¶5 1I strongly prefer all-in-one solutions, but the next best seems to be workflow solutions. 2A common one that some people I know use is InDesign for the page layout functions, numbering, and all annotations, and Illustrator for node boxes at whatever level makes you happy. 3This can work, but tends to be slower, and is prone to errors and problems with collaborative design due to the multi-part nature. 4There is a need to develop and adhere to a strict workflow.

3.4.3¶6 1Dedicated flow charting programs are an option for some tasks. 2Visio (and Omni Graffiti) have been stretched to do fairly good drawings, but I find them to be too limiting. 3Truly dedicated programs, like Rational Rose or other IDE-integrated diagramming tools are a thought, but are usually so very locked into their methods they are hard to use generically. 4The Rational products, for example, want to tie to their suite,

and export as code, so require a lot of work to do simple, user-centric design.

3.4.3¶7 1The problem with all of these flow chart programs is they are difficult or impossible to turn directly into detailed diagrams. 2The ideal program is something that can do flow charts with ease, and allow drawing of any shape you need for detailed work. 3With the open source movement especially we've actually considered and mapped out (twice in my career) simply making the perfect platform, but time and money always kill this.

#### 3.4.3.1 Versioning

3.4.3.1¶1 1This might seem like something really nerdy and IT-centric, but it's not for them, or the sake of process, but for good design reasons.

3.4.3.1¶2 1Okay, first I started because a version of Free-hand for a while liked to crash without warning, and cleverly ruined the file. 2You didn't just lose your changes since last save, but all your work on the whole project. 3So that was bad.

3.4.3.1¶3 1But something else came from it. 2Good ideas come and go, and drift in and out of the design. 3Now when you need a solution for a design, you can often just go back through every old version, look in the appropriate location, and see if any other options were considered in the past. 4It's best if you remember to write down everything, and why, but even without that you'll be surprised at how much information is left behind incidentally.

3.4.3.1¶4 1Version control software is not needed for this. 2If you are going to have more than one user on a single project at a time, be careful with workflow to make sure no one overwrites each other. 3Split the document into several documents, or branch it with each designer's initials on the end, or something similar.

3.4.3.1¶5 1Versioning for a single user is as simple as stopping, closing and opening a duplicate file every once in a while. 2I always add the date in a code to the file name, as well as some other stuff. For multiple versions on a day, I add letters to the end of it.

- 3.4.3.1¶6 1The third high level document created on February 14th would be labeled:

HL-client-projectName-021409b.fh1

- 3.4.3.1¶7 1You can use whatever system you want, but make sure everyone on your team understands it, so there is no confusion as to which is newest, and what abbreviations mean.

### 3.4.4 Types of Documents

- 3.4.4¶1 1All of the drawing methods described so far take place on a single giant sheet. 2If you can do this, that's tremendous. I had plotters at Sprint when I first started this, and could print my drawings 36" tall and as wide as I wanted. 3Sometimes over ten feet wide.

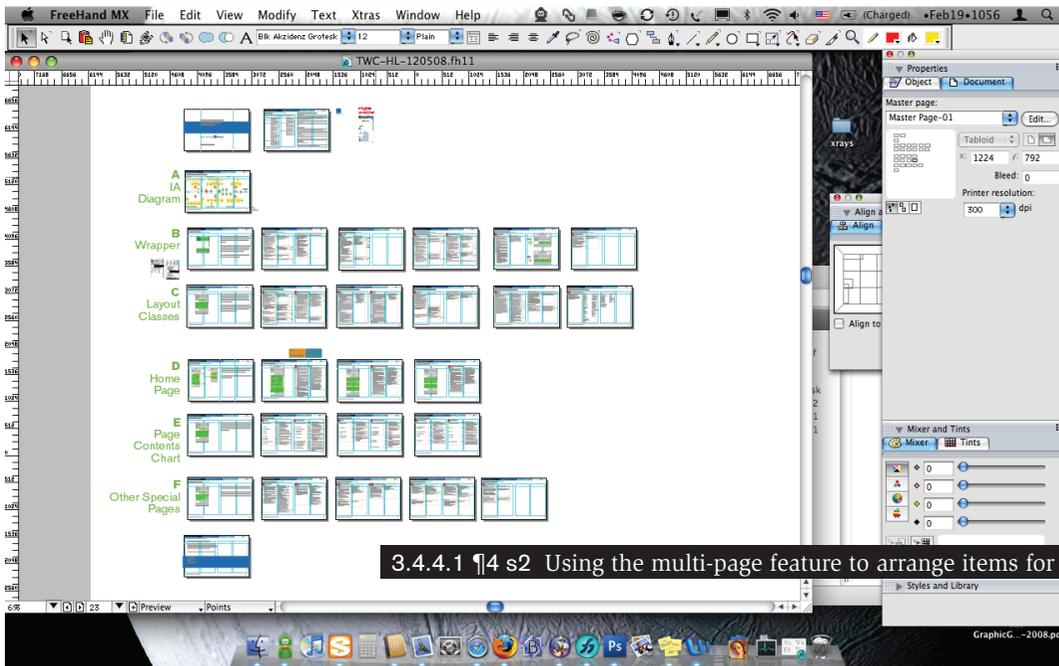
- 3.4.4¶2 1This also worked because everyone was in the same city, and often in the same building. 2I would draw, print, walk over to the meeting room and present the ideas in person. 3Clients and developers loved the giant drawings, and someone always took them back to their area to stick on a wall.

- 3.4.4¶3 1But that's about all that's good about it from a team point of view. 2No one seems to be able to zoom in and out in Acrobat, or any other program, so large drawings are hard for most people to use on their computer. 3For most teams now, whether they are inherently multi-national or simply as a result of outsourcing, not everyone is in the same area. 4And since practically no one has a plotter, they cannot make their own printed copies of such documents.

- 3.4.4¶4 1So, over time, I finally admitted defeat and do almost all of my drawings now in other ways. 2I still find drawing in the larger format to be useful, and encourage you to try that to start. 3Then, move the drawing to other formats when it is time to share with the project team.

#### 3.4.4.1 Other Drawing Solutions

- 3.4.4.1¶1 1When I was finally ordered to stop delivering giant documents, I surveyed every other project deliverable, the available printers and even looked over people's shoulders to see how they



viewed the documentation on screen.

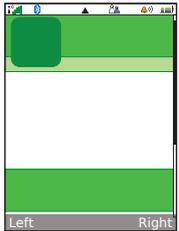
- 3.4.4.1¶2** 1This is largely how I discovered that no one knows that Acrobat can zoom, but I also found out a lot of offices have tabloid sized printers, and even when color printers are available, they are rarely used. 2So, I ended up making sure my documentation prints well in black and white, and settled on an 11x17” drawing size. 3Type size generally does not get too small, though, so it is just readable when printed at letter sizes. 4This has worked very well for something over 100 deliverables. 5For guidelines on type sizes and colors, see my sample documents.
- 3.4.4.1¶3** 1The other thing that has changed over time is my increasing reliance on object-oriented design. 2Though I am still sad about the loss of the integrated note and flow charting system, this re-usable component strategy meshes fairly well with a page-by-page specification style.
- 3.4.4.1¶4** 1A key to make this all work is to keep the principles in the UML-style modeling, with sections and strict numbering. 2When drawing in Freehand, which lets you put pages wherever you want; I still work in a system reminiscent of the one-page view, with each section’s pages grouped, and labeled so the label can be seen

from a distance.

- 3.4.4.1 ¶5 1 Because of the lack of such grounding by sections for the page-by-page view when exported, create an introductory page for each section.
- 2 The emphasis here is on the title of the section, and a summary in text of what that section does and how it meets the overall design goals of the project.

The Weather Channel Mobile Web 3.0
High Level Design Document
December 5, 2008
Version 120508
little springs design

## B Page Wrapper



The wrapper is the set of elements that surround the content on a set of pages. It is not the template, but items that are reusable between individual pages and page templates.

For this design, the page wrapper is essentially identical on all pages, including the home page for the site, and basically encompasses the header, footer and certain localization elements that launch from there.

Notable variations do exist between device classes, to which careful attention must be paid.

©2008 Little Springs Design - Proprietary & Confidential

3.4.4.1 ¶6 A tabloid diagram, with introduction to the section, followed by individual node details

The Weather Channel Mobile Web 3.0
High Level Design Document
December 5, 2008
Version 120508
little springs design

## B Page Wrapper > ii

A1 Masthead	
<b>Information:</b> <ul style="list-style-type: none"> <li>The Weather Channel logotype</li> <li>A background, controlled by content management to change by seasons or other occasions</li> <li>Current conditions icon: display</li> <li>Current temperature in the user selected unit</li> <li>For any device with a keypad, display the (1) acceskey</li> </ul>	<b>Behaviors &amp; functions:</b> <ul style="list-style-type: none"> <li>Any device with a keypad (1) to access current conditions</li> <li>Link to advertising</li> </ul>
<b>a - Best-case devices</b>	
<ul style="list-style-type: none"> <li>Icon indicating settings</li> <li>Anywhere but home page:                             <ul style="list-style-type: none"> <li>Icon indicating home page</li> <li>Print name and zip code of current location</li> </ul> </li> <li>Indicator for change or more information adjacent to this</li> </ul>	<ul style="list-style-type: none"> <li>[Settings icon] link → C2 (7)</li> <li>Anywhere but home page:                             <ul style="list-style-type: none"> <li>[Home icon] links to home → D</li> <li>&lt;location&gt; link opens layer → A4-a</li> </ul> </li> </ul>
<b>b - Dynamic scripting</b>	
<ul style="list-style-type: none"> <li>Print name and zip code of current location within a selector</li> </ul>	<ul style="list-style-type: none"> <li>Selector for location, detailed in A4 which see</li> </ul>
<b>c - Non-scripting</b>	
<ul style="list-style-type: none"> <li>Print name of current location</li> </ul>	<ul style="list-style-type: none"> <li>&lt;location&gt; link opens new page → A4-c</li> </ul>

**d - Pages without location**

- For certain pages, local conditions may be confusing, irrelevant or not available and are not shown
- In the masthead, suppress the following information:
  - Alerts
  - Local weather conditions icon
  - Local current temperature
  - Accesskey to view current conditions
- Localization selector
- This applies to the following pages:
  - 2.2, 2.2.1 - Storm/Hurricane
  - 2.3.1, 5 - Regional weather details
    - 3.1 - Airport delays
    - 3.2 - Airport delay map
    - 4.5 - Snow & ski
    - 4.7 - Boat & beach
    - 5.5 - Celebration weather
    - 6.5 - Celebration weather
    - 7.1.2 - Change location
    - 7.1.2.1 - Localization interceptor
    - 7.2 - Feedback
    - 7.3 - Registration interceptor
    - 7.5 - Message page

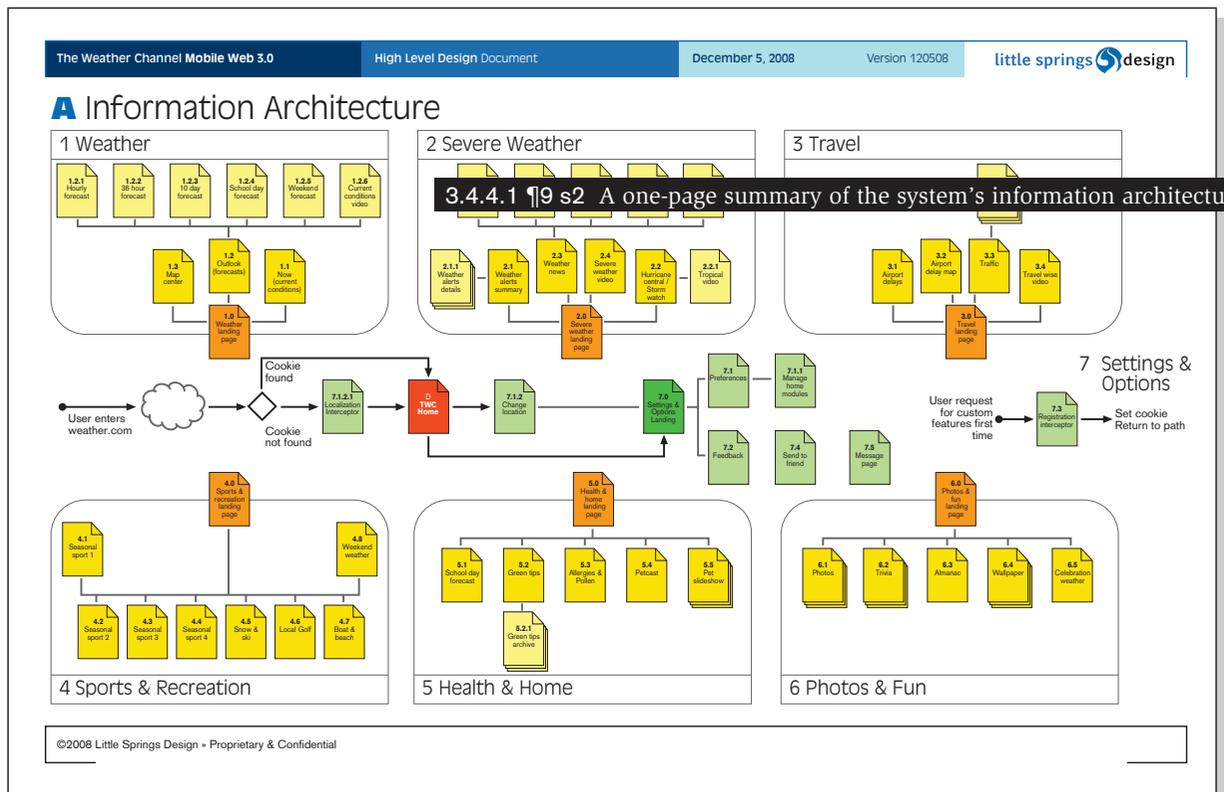
- No new functions in this module

©2008 Little Springs Design - Proprietary & Confidential

3.4.4.1¶6 1 Sometimes, as shown above, this should actually have diagrams of how it relates to the concept documentation. 2 The example shows page diagrams, as this section is about modules that exist on each page of a certain type. 3 When pages or states are shown within an entire system, the location within the overall architecture can be shown instead as a snippet or highlight of a flow chart.

3.4.4.1¶7 1 Aside from the loss of the neat single view of the whole system, there is also simply no flow at all with such a document. 2 Each interaction needs to have an arrow leaving the node (shown in green on the illustration) referencing the element loaded, or looping back to show that it reloads the current node.

3.4.4.1¶8 1 Another item must be added to the drawing, which is the overall flow, or information architecture. 2 I have always enjoyed a system of explaining the whole process, then showing in great detail the flow by sections. 3 However, this



has fallen by the wayside as another item that is used as a snapshot or executive summary.

3.4.4.1¶9 1Although you can get away with a summary one-page IA, followed up with detailed, multi-page flow charts, this is extra work, can become out of synch and I fear it could lead to confusion on the development teams. 2So I simply compress the entire system diagram onto a single page.

3.4.4.1¶10 1There are occasional cheats, such as the stacks of pages icon to denote more than one item that cannot be drawn, but generally it is possible to get the entire system onto a single page. 2Use grouping boxes, which must reflect the groups in the rest of the diagram, number all elements to refer to documented nodes, and consider things such as color to separate types of content, sections or functions.

3.4.4.1¶11 1And feel free to do what makes the most sense and break out of the strict style sometimes. 2If there is a very complex process, such as most signon systems, feel free to summarize here, and detail it somewhere else. 3Just consider all the issues and failure points to assure it works for everyone on the project team.

#### 3.4.4.2 **Distribution and Delivery**

3.4.4.2¶1 1The high level diagram is not going to be beloved by many on the project team. 2It's boring and full of words instead of pretty pictures. 2But insist it's an important deliverable, get it on the project schedule, and present it to everyone.

3.4.4.2¶2 1Everything else about presenting these documents is the same as for other documents, so see the presentation and delivery section later.

### 3.5 Detailed Interface Designs

- 3.5.1 1It's easy to think this document is the culmination of the design process, and most clients and implementation teams will believe this.
- 3.5.2 1But I urge you to not think that at all. 2The process as a whole is key to creating this document, and when delivered you have to assure the actual product is built to those specifications.
- 3.5.3 1So, if you have jumped to this section to find out how to create detailed designs and specifications, stop that. 2Go to the beginning, and read the whole book.

#### 3.5.1 Naming and Understanding

- 3.5.1.1 1I spend a lot of time on semantics and naming items to make them as clear as possible. 2Yet for this one, I am a bit uncertain. 3As originally developed, it rapidly became the UEAW, for User Experience Annotated Wireframe. 4The UE part was the team name, before we knew that Xs were cooler, and the annotation was clear at that point.
- 3.5.1.2 1Now I tend to only use "wireframe" as a descriptive, and push the concept of a specification. 2Hence, it's the generic "Detailed Design Document" or similar. 3I have decided here, where we're having this discussion, to use the "interface" portion as well. 4While the document is specifying interaction, it's the same interaction you already drew in the high level document. 5What you now work on is building the interface elements that support that interaction.

#### 3.5.2 Specifying and Drawing

- 3.5.2.1 1The detailed design is a model of the system. 2You can think of it as a sketch or refer back to the origins of the word "wireframe." 3The roughed in model you can pick up in your hands (or walk through on the computer). 4With this rough model you can get an idea how it relates to itself, and its environment, make sure there are no sharp edges sticking out, or pieces that cross each other in impossible ways.

3.5.2¶2 1It's also a specification. 2A document detailed, and specific enough that the implementation teams can understand what to build, even if they are not as good at visualizing things. 3Specifications are in the language of the craftsmen, assembly line personnel and developers.

3.5.2¶3 1Do make sure it's only detailed enough. 2Try to restrain yourself from overdoing it. 3A 200 page document is hard to carry around and flip through; it is likely to be ignored due to inconvenience.

### 3.5.2.1 Details and Documents

3.5.2.1¶1 1As a general rule, if you wrote or drew something in the high level document, you will be moving it to the detailed document, and adding more detail.

3.5.2.1¶2 1But the exceptions to this are key to understanding it. 2The high level document is not just a draft of the detailed specification. 3Especially for certain documentation methods, many details of the flow, or system interaction will not be suitable for the detailed wieframe. 4They can be referred to, but left in the high level document.

3.5.2.1¶3 1Additionally, the manner of documentation will change. 2There should be very few or no notes directly on the drawings. 3The operation of the system should be directly indicated by the drawings, and the labels of each element. 4Detailed specifications will be in annotations adjacent to the drawings, in another document, or even simply refer to the existing high level document.

3.5.2.1¶4 1References between the documents should be easy, as the numbering scheme should not change. 2For any element from the high level document, use the same number and title on the detailed document. 3Never re-use a number, even if you delete it. 4Annotations, notes and issues should follow and refer to this scheme as well.

### 3.5.2.2 From Words to Boxes to Pictures

3.5.2.2¶1 1Moving from the high level to a detailed document should, in a perfect world, be a snap.

2Occasionally I have seen it flow easily, and not just by me but by people who did not go to art or design school.

3.5.2.2¶2 1However, all of these were for Sprint, or some other client for whom there was an existing relationship and an established set of guidelines about how to draw the pages. 2If there's a style guide, or enough extant design that you can make one up by spending some time looking at their products, follow their lead.

3.5.2.2¶3 1For all other situations, I'll admit it can be a little harrowing. 2But you can improve this by reverting back to some process tricks, and carefully following the design principles for each and every decision.

3.5.2.2¶4 1The high level document will be your guide, or your outline. 2I pretty much always start a detailed document by simply making a copy of the high level document and re-naming it. 3Then I go through and replace each high level module with a detailed diagram, being sure to represent each and every item listed on the high level boxes.

3.5.2.2¶5 1But this, on its face, seems nonsensical. 2Lists of information, sorted by type, are not a visual design. 3Right. 4But they have design information. 5Go back and review the discussions of information design theory, of the creation of box models, and of the hierarchy of communication:

Position > Size > Shape > Contrast > Color > Form

3.5.2.2¶6 1You are likely, in fact, to have some of this information laying around. 2Whether it is sketches, or post-it boards or concept documents and explanatory decks, pull them out and use those as your baseline design. 3Look at the information you have within each screen and on-screen element for a state, and re-apply the information design process to assure the correct items are emphasized.

3.5.2.2¶7 **Filter**

- 1Hide all information, process or system interaction the use will not need.
- 2Make visible all information, processes or tasks the user will want or need.
- 3Define the limits of any user interactions or

3.5.2.2¶4  
This hierarchy of communication is essentially of my own devising. It's derived from similar lists by any number of design theorists. These are fundamentals that do not explicitly take into account interaction, and waiting, space, time, context and other concepts, but whenever I try to add those it it just confuses the issue. So this is my list. You may disregard it, but find something to use as an organizing principle.

customizations.

#### 3.5.2.2¶8 **Group**

- 1Determine user goals and tasks within the remaining information & functions.
- 2Group information and functions by task and goal.
- 3Group all other information by user logic.

#### 3.5.2.2¶9 **Prioritize**

- 1Organize groupings heirarchically by user interaction, user logic or task process flow.
- 2Offer the lowest number of user interactions (clicks) possible.
- 3Consider architectures that present multi-layer information in an apparently flat manner.

#### 3.5.2.2¶10 **Arrange**

- 1Place items on individual pages based on grouping and priority.
- 2Specify a single template for use on most or all pages.

#### 3.5.2.2¶11 **Optimize**

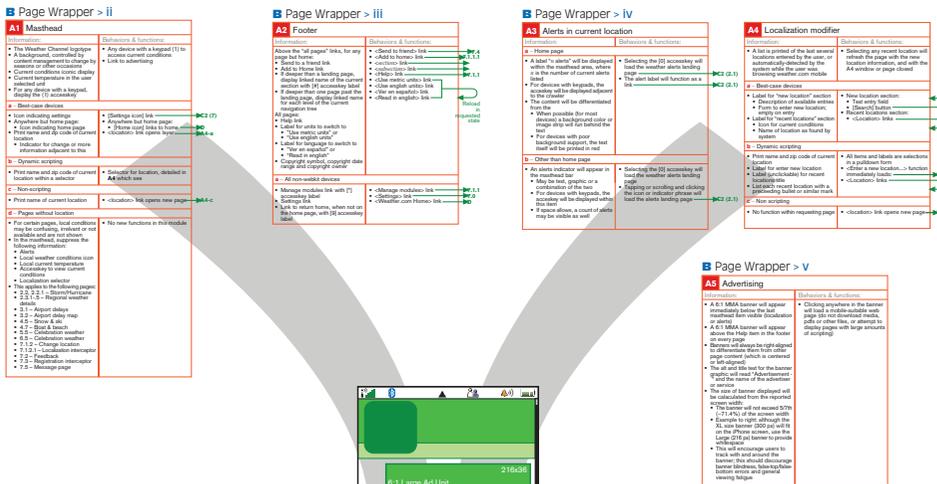
- 1Revise the design, iteratively and with user feedback if time allows, to assure it is the right solution.
- 2Confirm with technical resources that it can be built and modify as necessary if not.

#### 3.5.2.2¶13

Regarding object-oriented design and my comment about a “one time style guide,” you can sometimes make this a more-than-one-time style guide. If there are more projects within the product or brand, or might be, make these styles explicit and follow them in the future to save even more time.

3.5.2.2¶12 1It is, of course, rare that you would eliminate something at this step of the process, but there are design solutions that allow information or functions to be hidden-by-default (window-shades, floats, popups, etc.). 2This process can help you discover those needs. 3It is also always a good idea to review your work, so there’s nothing bad about re-applying the whole process this late, time permitting.

3.5.2.2¶13 1Object-oriented design might seem to make this whole process more complicated, especially the portions on polymorphism and so forth that I discussed above. 2But really, it makes it a snap. 3Because instead of designing each and every part, you are creating a sort of one-time style guide. 4Besides using the same element for



3.5.2.2 ¶6 Use the info design process to turn high level specifications into drawings

**Location**  
The masthead will be of the default, location-displaying type.

**Title**  
Explain the purpose of the page with the title. No introductory copy is necessary.

**Sections**  
The title for each selection set should serve, without description, to explain the choice available.

**Examples**  
When an option may be unclear, provide a sample of the data that will be shown.

**Large lists**  
For items over three selections long, use a pull-down list instead of a series of radio buttons.

**Buttons**  
Functions are provided to both submit (save) and cancel or back out of this page.

Cancel will go back one page in the history, to the previous page (probably 7.0).

**Cookies**  
Explain that the settings will be saved as a cookie for subsequent sessions.

repeated use of search, or contact, or the navigation, you can re-use the key attributes of the design. 5A form for settings can almost immediately become a form for contact information.

3.5.2.2¶14 1It's not cheating to re-use elements, especially your own. 2And within a single design it's the only real way to assure consistency.

3.5.2.2¶15 1In my ideal world, the document moves directly from high level to detailed, with all of the flow lines and decision points still in place. 2This is also terribly easy to explain, but runs into any number of problems with comprehension by the

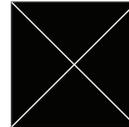
consuming teams, and delivery issues with large documents. **3**Formatting of the document as a whole, and distribution, will be discussed a little later on.

- 3.5.2.2¶16** **1**The high level document will be referred to, and will guide you, but also will almost certainly need to change. **2**Much as discussed above for concept documents, expanding on design, improving and changing it at the detail level will affect the high level document, which must then be updated. **3**The strategy I referred to of moving straight from high level to detailed design makes this easy; if you change or add something, just go to the high level document, and add it in at that moment.
- 3.5.2.2¶17** **1**Unless it makes no sense. **2**Many design decisions you might make at the detailed level will suddenly make no sense when you try to add them to the high level document. **3**Some are easy, and you will simply be reminded of a contrary decision or conceptual foundation. **4**That is why referring to the previous document is so key to good design.
- 3.5.2.2¶18** **1**But sometimes changes are simply difficult to fit in properly. **2**Follow the lead of the document here; if it's hard to add in, there might be a reason and you are at risk of violating some core principle of the design you may not have yet recognized.
- 3.5.2.2¶19** **1**Naturally, by the time you get to this level, you also better be not just versioning your documents, but keeping a change log. **2**Don't just add in the changes, but write down what you changed. **3**This is, aside from helping the implementation teams directly, a useful cheat sheet for you when presenting changes and conducting further revisions.

### **3.5.3 Content, Brand and Style**

- 3.5.3¶1** **1**Detailed documents must by their nature represent content. **2**Many wireframe techniques use boxes or other greeking methods. **3**I favor going as far to the final solution as possible.
- 3.5.3¶2** **1**There are several reasons for this. **2**First of all is that greeking often confuses the project team.

Title



Title



Title



Title

Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est.



### Designing with words

I especially agree that a key portion of why I hire someone (when I did the hiring) was if they can express an awareness of why they made design decisions. I have hired a lot of people whose personal art or design style I despise; but they have one, and are conscious of it enough they can pursue any design they need.



#### 3.5.3 ¶1 s3 The many levels of greeking, and depicting final content, which I suggest

3They do not have an innate sense of what a box or lines to represent type mean. 4There is a significant likelihood of fake text being implemented. 5I have personally been involved in several cases where this happened, and grew tired of having to explain why there are jokes or inappropriate phone numbers in production.

3.5.3¶3 1There are also advantages in selling the design as final, and understanding of the design by client teams. But most of all, it is a chance to extend design into the content area. 2You can work with content creation teams to make the text or graphics fit in the interactive design.

3.5.3¶4 1Very often, content will be created for your project. 2Try to find out, as early as possible, who will be creating it. 3All too often, you might be expected to do so, but this is difficult even if your team has writers, as you will not know the product as well as the client group. 4Try to work with them to get at least basic information, and

work through making it readable, and making it fit into the available space. **5**If there is a dedicated writing team of any sort, you can even provide them minimal training on the wireframe process, and give them access to the documents; any content you need created can be directly entered into the design document.

**3.5.3|5** **1**As you are a design professional, writers have a specific skillset that cannot be simulated. **2**I rely on writers whenever possible to help me create everything from page titles to button labels. **3**They will do a better job, in less space, than you can do (usually). **4**Graphics are similar, and a good illustrator, photographer or graphic artists will care how and where the graphics will be used and will be happy to work with you in this.

**3.5.3|6** **1**In the opposite case, fitting content is even more important. **2**If there is an existing data store, say of customer and billing information, you cannot change it. **3**Using fake content will help no one, and your design will not serve the information or the user well.

**3.5.3|7** **1**Content will, often, still be placeholder as the final is not available. **2**At the least, it will be merely representative of the numerous states the information can take on. **3**Be sure to explore minimum and maximum lengths, to be sure no issues will arise. **4**Use very long names, large numbers, and different dates (depending on your information) to make sure it flows correctly. **5**Do not forget to check for undersized items, especially graphics, as well. **6**Often, designs look strange with a single line of text, or an icon-sized graphic centered on the page. **7**Revise the design, or develop a method to set standards for content to avoid troublesome content.

### **3.5.4 Technologies for Drawing**

**3.5.4|1** **1**I will try not to repeat myself too much here, which also means you need to look at previous sections to get my complete thoughts on this topic.

**3.5.4|2** **1**Drawing programs, for example, I am not going to discuss. **2**Except to say this transition from one diagram to the other is exactly why I favor

a drawing program, and not a flow-charting or diagramming program for the high level documents. **3**The shift between programs can cause a mental disconnect, regardless of your intended procedures.

**3.5.4¶3** **1**Here, even more than in the high level document, you must version carefully and regularly. You should be playing with the design a lot more, and concepts will come and go in a few minutes. **2**Try to save new versions regularly, or at least remember to push those ideas off to the side for later reference.

#### **3.5.4.1 Scale**

**3.5.4.1¶1** **1**One of the more critical, and least obvious, tactics that I use for drawing is to have a scale. **2**The way Freehand works, this usually is 1pt = 1px. **3**Any scale will do, however it pretty much has to represent the design in pixels if you are doing anything interactive.

**3.5.4.1¶2** **1**Why? **2**Because the detailed wireframe is specifying, and expressly stating design attributes, not just implying things about the interaction. **3**Building to scale, and using the right typeface at the right size, gets you very close to the final design.

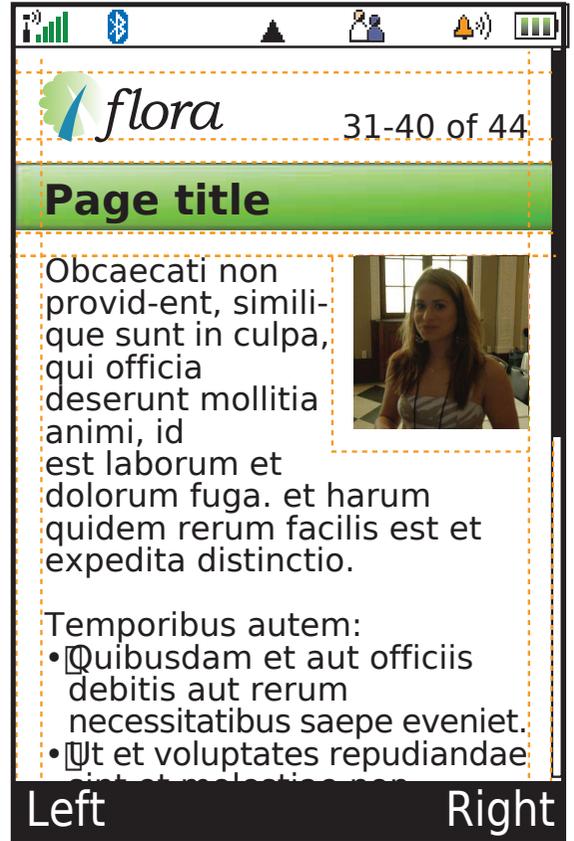
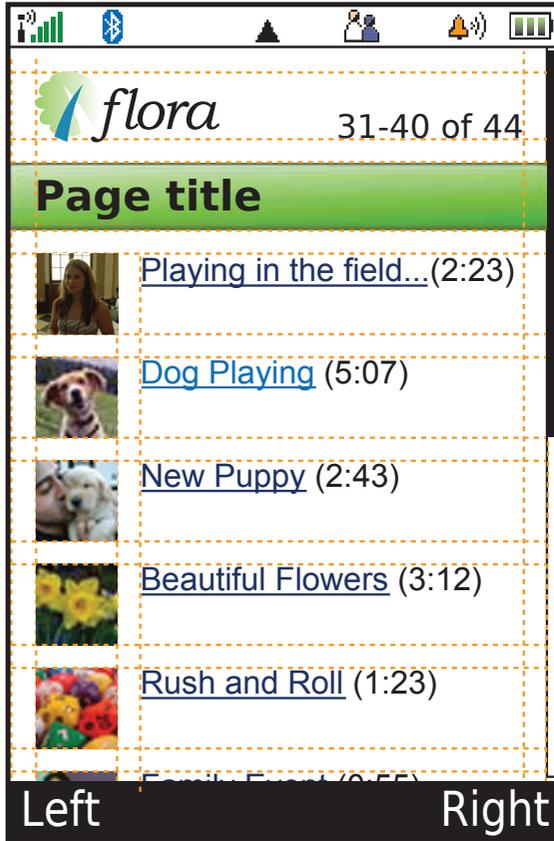
**3.5.4.1¶3** **1**Many people, seeing this, wonder why I don't use a raster design tool. **2**There is in fact a rapidly emerging subset of designers who use Fireworks for just this reason. **3**I feel, after playing with it a bit, that's they are on the wrong track.

**3.5.4.1¶4** **1**Some of the issues I have encountered are perhaps minor or could be overcome. **2**It is harder to enter and edit type, and add items. **3**They are inherently one-page, unless synched with a page layout program. Things like this slow you down, and tend to mean that the designer is the only one who can touch the drawing. **4**See the above section on content and working with writers.

**3.5.4.1¶5** **1**But most of all they are inherently raster graphics. **2**The design elements loose all sense of portability, and variable scale, when locked into a raster-output program. **3**And there's a key attribute: It's a scale, not a size. **4**A single design can rapidly be moved to several different screen

# 177px or larger

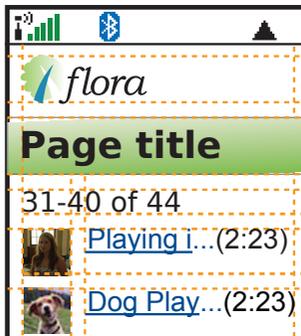
Shown on 240px wide screen



3.5.4.1 ¶1 Using a scale allows clear depiction of the final state, and of the relationship between screens

# 176px or smaller

Shown on 128px wide screen



sizes (or tried out in different ways) because it is vector based, and because the scale is a known elements. **5**Raster solutions to this instead set a size for the entire screen. **6**While simply a bad precedent for most desktop computer applications and websites, this is a disaster for other design cases, like mobile phones.

#### 3.5.4.2 **Type, Color & Graphics**

- 3.5.4.2¶1 **1**Especially when making wireframes that are supposed to reflect reality, everything else must begin to approach full scale and imply the appropriate behaviors.
- 3.5.4.2¶2 **1**Use the correct type (fonts) at the right sizes and weights. **2**If you use something else, it will take up a totally different amount of room, which will make much of your design (and much of that help from the content team) moot.
- 3.5.4.2¶3 **1**Color and imagery are probably the two biggest arguments I get into with other types of wireframe aficionados. **2**Most insist they have to be black and white. **3**A lot of those actually mean grayscale, and I can get behind that.
- 3.5.4.2¶4 **1**But there is also value in color, and actual graphics. **2**Sometimes, it is critical that actual (or placeholder) images or banners be used. **3**The way they interact with the design is very different than a big, black box. **4**Often, logos are type-based, without a defined edge. **5**An outlined or grayscale version will often work, but a black box will misrepresent it, and make design difficult.
- 3.5.4.2¶5 **1**As far as color, I almost always make links in color, as well as underlined; even if the color is not that which will be used in production, or the designers insist on no-underlines (a whole separate argument) it helps explain where the linked items are.
- 3.5.4.2¶6 **1**Other color treatments have to do with how much you need to communicate, and how hard it is to sell the project team on the design. **2**While I often stick with grayscale (but otherwise realistic) designs, I will add color when it is key to the brand, is the first product, or there are communications difficulties. **3**I might be cheat-

ing and playing off the client's emotions, but if it gets the project out the door in good condition, my morals are fine with it.

### 3.5.4.3 Templates and Guides

3.5.4.3¶1 1Parallel to the discussion of style guides is the use of templates. 2There are many out there for design of desktop websites and a few for mobiles. 3But unless required to comply with an existing structure, you will need to decide what is right to use for your design and what matches the brand.

3.5.4.3¶2 1You are also likely to make this up as you design. 2A few pages in, when you start feeling comfortable, define a template or grid based on the margins and other spaces you have created. 3They won't all be the same, so make decisions and stick to them.

3.5.4.3¶3 1Depending on your drawing tool, you can use the guides built into the program, or may want to draw guidelines by hand and copy them to each page template. 2Then use tools inside the drawing program to align items to the guides. 3Eyeballing will never be as good or as accurate as mathematical precision. 4Your teachers were right; you will use math every day.

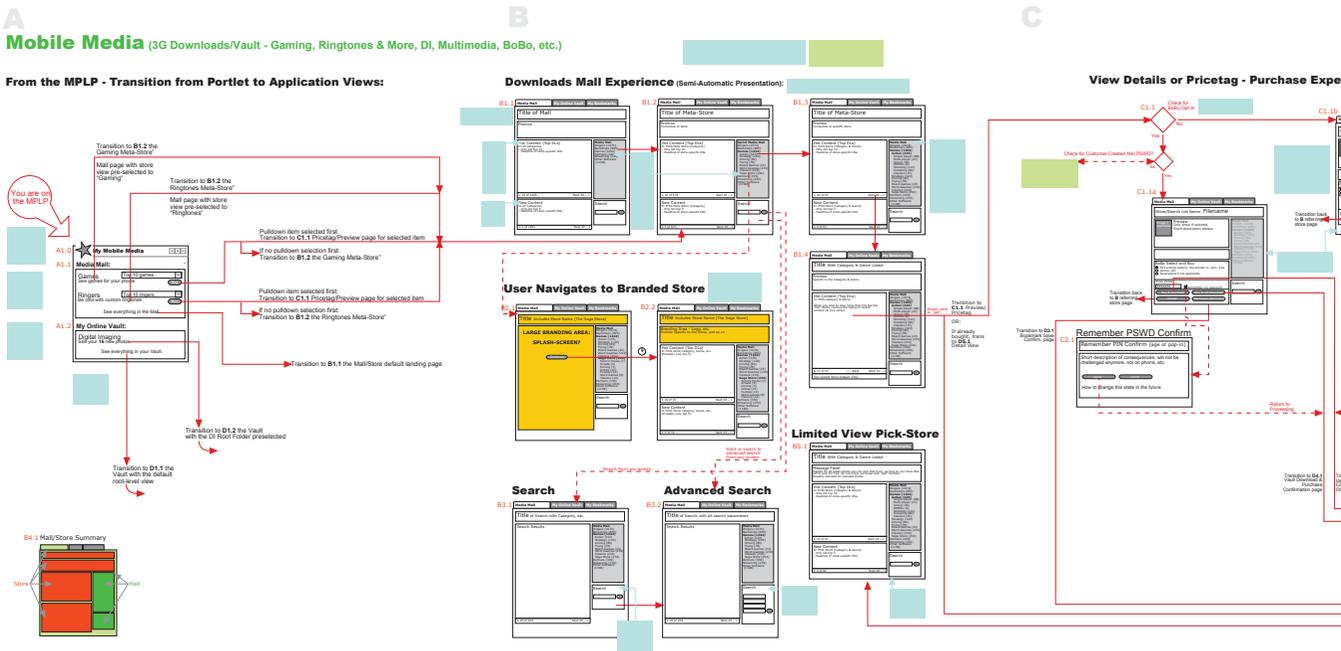
3.5.4.3¶4 1Use some caution in selecting color and pattern for the guidelines if they will be visible in the final deliverable; I have had developers ask how to code the little blue lines they saw all over my wireframes. 2Twice.

### 3.5.5 Documentation

3.5.5¶1 1Another key aspect of the detailed design document is that it is a document. 2The ability to distribute it, and for consuming organizations to understand and want to use it, have made a huge impact on the style of the final deliverable over the years.

#### 3.5.5.1 Big Wireframes

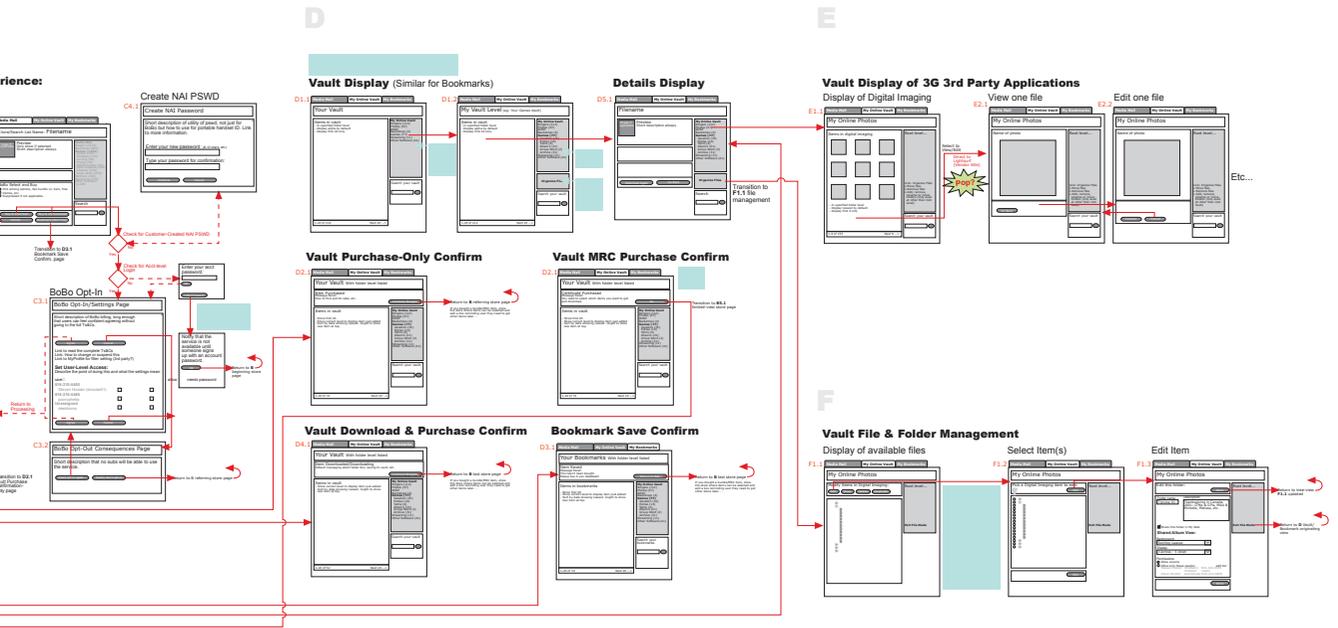
3.5.5.1¶1 1I used to have one, perfect deliverable. 2A single, huge flow chart where each node is a detailed (correct to scale) drawing of the interface



for that screen or state. **3**Flow lines came directly from links and moved directly to the next state depicted. **4**Notes appear in boxes adjacent to each element, at decision points, and in the corner of the page.

**3.5.5.1¶2** **1**This is a single document. **2**All design deliverables in one, and all older documents are abandoned (unlike what I just said about referring back to the high level document). **3**It's impressive, and if you have a plotter and can walk the drawings over to people, clients and developers enjoy hanging them on their walls.

**3.5.5.1¶3** **1**As long as you have a chance to walk people through it, they also are pretty easy to understand. **2**But everyone tends to want a printout, and electronic distribution does not go well. **3**Delivery to remote teams will involve printing and mailing, and still may fail.



3.5.5.1 ¶2 The single, large wireframe with all annotation, flow and page details in one location

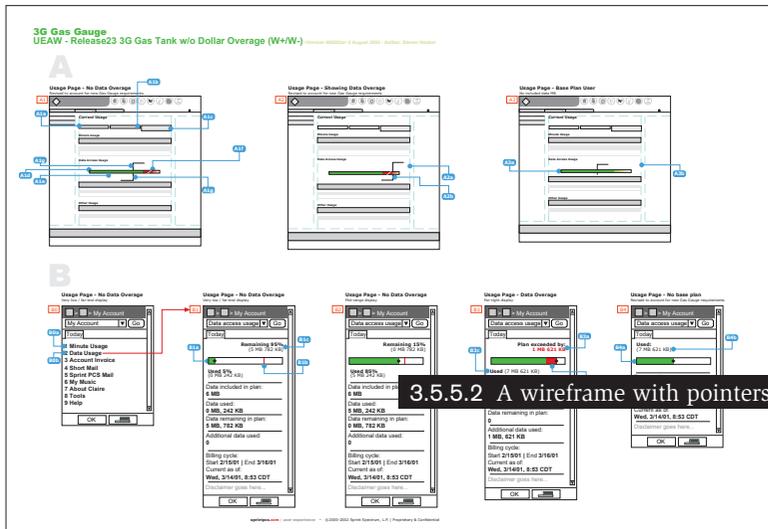
### 3.5.5.2 Annotated Wireframes

3.5.5.2¶1 The “big wireframe” was developed at a time when all UX folks (at least the ones I knew) were being more formally integrated into design processes (and IT processes). 2Artifacts were created to accompany this, mostly derived from systems specification styles.

3.5.5.2¶2 1This meant that for a very long time I was creating specialized design documents, and then cutting them apart to go inside Word documents. 2One way to try to start addressing this was to remove all the notes from inside the wireframe. 3Instead, there were numbered pointers.

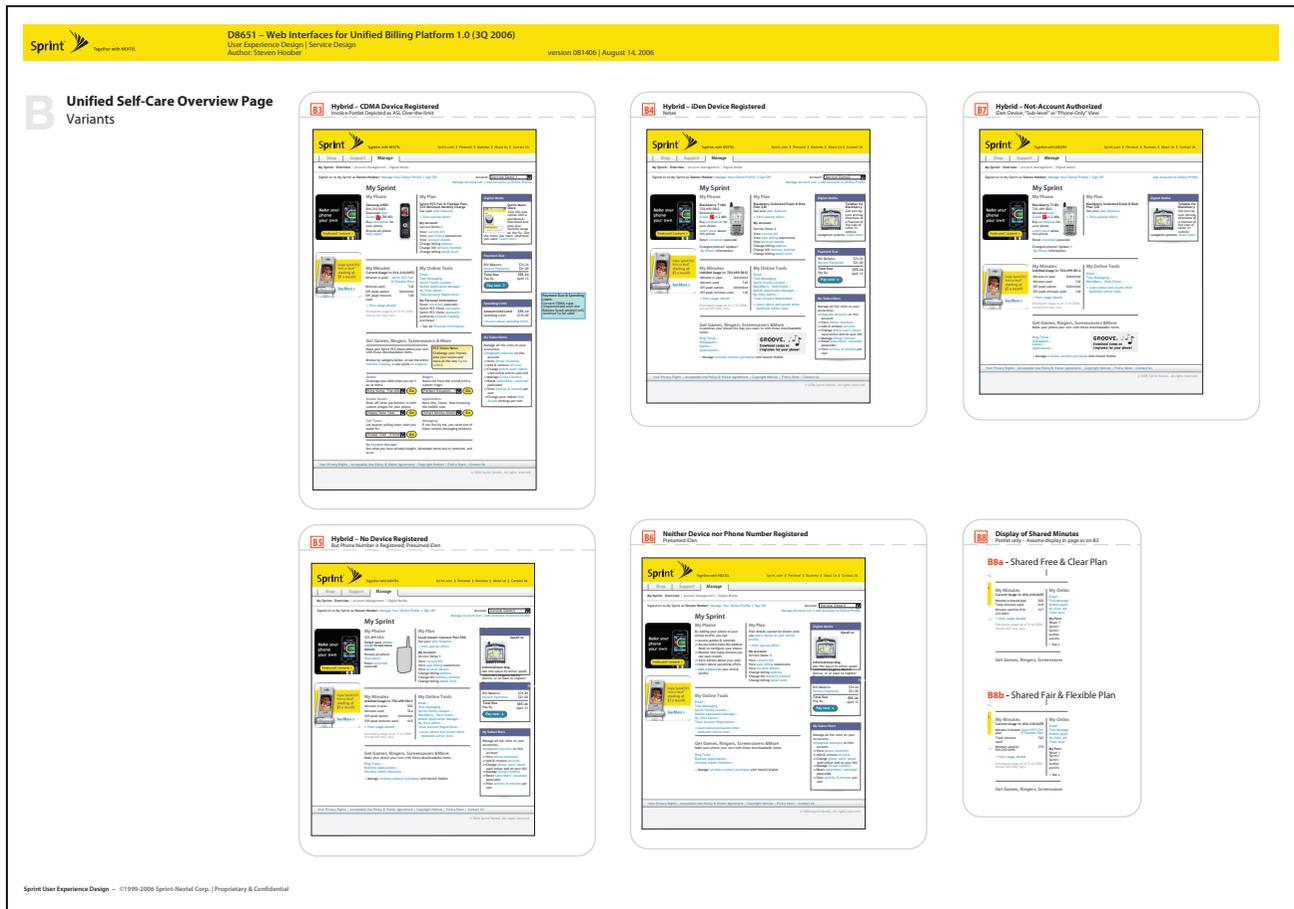
3.5.5.2¶3 1The text specification still existed, but was now integrally tied to the wireframe.

3.5.5.2¶4 1Though this did lighten the wireframe (by removing all the boxes) and created a more formal specification out of the notes, it was ultimately not successful. 2Neither document was useful



3G Gas Gauge		UC Wireframe - Release-07> 3G Gas Tank (W+WW) 10 July 2002	
Navigation			
<ul style="list-style-type: none"> <li>Users click on the Current Usage Link under My Plan icon (W+)</li> <li>Users click on the Detail button on My Current Usage page on the landing page (W+)</li> <li>User selects My Account to access links to usage data (W+)</li> </ul>			
Annotation	Description	Comments	
A1	Account Level Information: A1a: Start - beginning of billing cycle. A1b: End - end of billing cycle. A1c: Current as of - day, date, time of most recent data pull.	Each level of display is present in three categories of information for (W+): Minutes Usage, Data Access Usage, and Other Usage.	
A1	Data Access Usage Section (No Data Coverage) A1a: Gauge shows total plan amount plus a generic coverage area. A1b: "In Plan" is shown in Green area and is between 0% to 100% of plan; pointer moves relative to amount of data used. A1c: "Overage" is shown in the Red area, however the pointer remains fixed once the coverage condition is reached. Break in bar indicates confusion as if it does faded not seen at right of bar. A1d: Indicate display percentage and amount of data used as well as percentage and amount of data remaining when in Plan.	Current usage pointer on other page (A1) moves from left to indicate total amount of data used up to 100% of plan. When in an overage condition (A1) pointer remains fixed in Red area.	
A2	Data Access Usage Section (Data Coverage) A2a: When in coverage condition show amount of data coverage. A2b: Display total amount of data used (plan plus overage) without showing percentage.		
A3	Data Access Usage Section (Basic Plan User) A3a: Current usage is shown in Green area. A Red overage area does not exist because a true "overage" condition does not exist. A3b: Indicate display percentage and amount of data used.		
B0	My Account B0a: Minutes Usage (minutes) Amount (change) displays relative use for each applicable category (Anytime, CD/Plan, Streaming) on a single page. B0b: Data Usage (new selection) provides user with access to gas gauge and data usage in 3-dim.	Pointer movement will be restricted on handset and will show relative usage.	
B1	No Data Coverage B1a: Gauge shows total plan amount plus a generic coverage area. B1b: "In Plan" usage is shown as a Green area and is between 0% to 100% of plan; pointer moves from left to right as data are used. B1c: "Overage" is shown in the Red area of the bar. B1d: Indicate display percentage and amount of data used as well as percentage and amount of data remaining when in Plan.		
B2	Data Coverage B2a: When in coverage condition, display the amount of data coverage in place of remaining. B2b: Coverage is shown in Red in the right of the bar.		

3.5.5.2 A wireframe with pointers and annotations as a separate text document

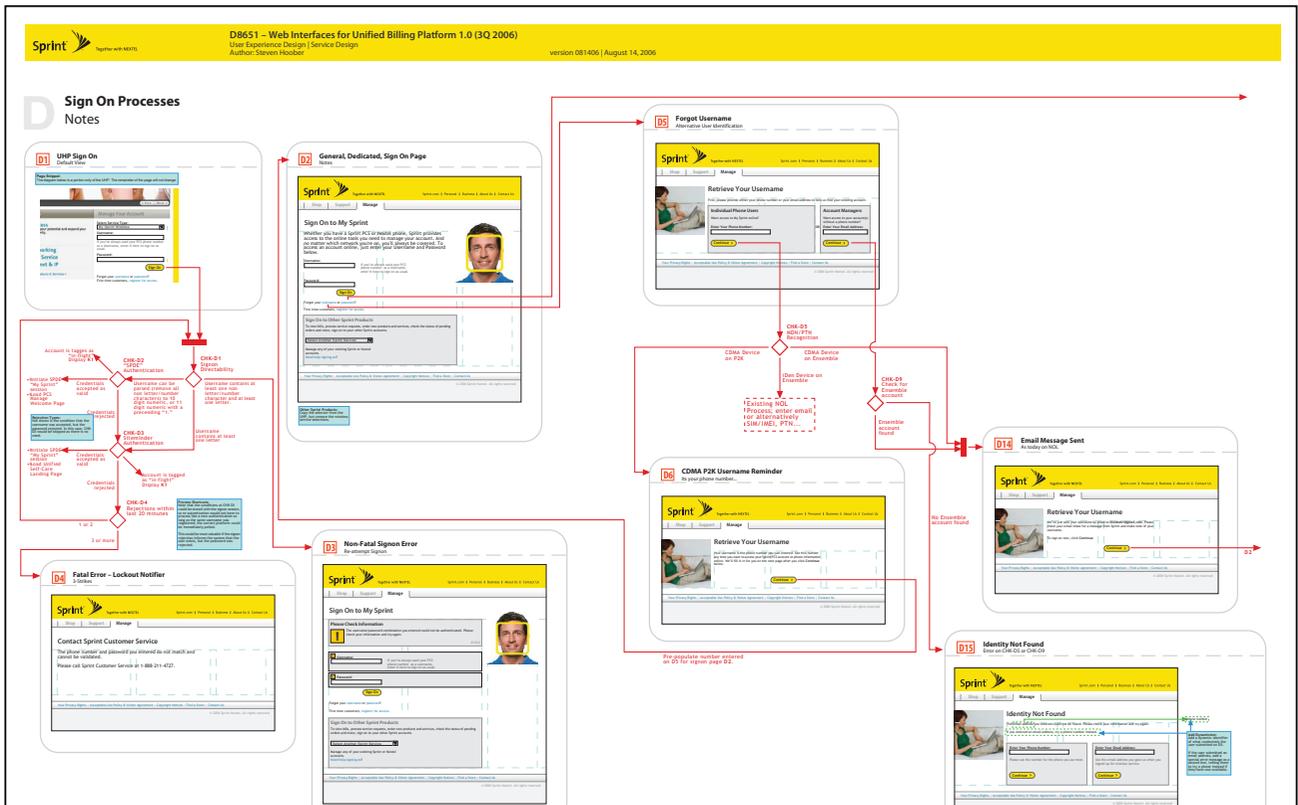


on its own, so features were missed or misunderstood, and since only the text document was perceived as readable on a computer, it was distributed and marked up without the wireframe. **3**The result was generally to include images of screens in the specification, ruining the concept.

### 3.5.5.3 Multi-page Wireframes

**1**When working with remote teams, there is really no way to avoid having to make multi-page wireframes instead. **2**This, very sadly, leaves behind the whole concept of a single flow through the interactive process, but as long as the principles are adhered to during the design, it can be made to work very well.

**1**The most basic version is to simply split up the diagram. **2**Some of these are easy, as they are free-standing landing pages. **3**Others can be



3.5.5.3 Two pages from a typical multi-page wireframe – Each section is a complete diagram

arranged so a single section (a letter element by the high level parlance) occupies a page. 4Sections that are too large will simply need to be broken up, or arranged across two pages with flow lines running between them.

3.5.5.3¶3 1Any notes are embedded within the document, either as boxes, or as entire sections explaining how a particular function or subsystem works.

3.5.5.3¶4 1This works okay, really. 2It's the purest form of wireframing as I like it on printable pages (though I never go under 11x17" paper). 3But it is graphically intensive, and multi-page. Some client have been a little baffled, and some

#### 3.5.5.4 Specification Documents

3.5.5.4¶1 1I actually use all of these still. 2But the styles I most often use are yet another multi-page version. 3I do prefer many other deliverables for the assistance they give to the design process. 4However, I became aware that a key factor in almost all projects that had failed to meet their goals was that the design was not executed correctly.

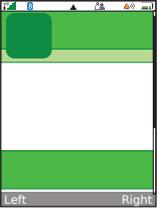
3.5.5.4¶2 1Communicating, presenting or even selling the design to the client teams, the developers and other implementation teams, and system test, was crucial to the success of the project. 2With these teams on board, or key members like test to argue for your design, it would get built much more easily.

3.5.5.4¶3 1And so, I adopted some ideas from books, from design magazines and from some other types of deliverables. 2I made a design presentation. 3Each section is organized to have an introduction, including design objectives. 4Each page of the document outlines a single concept (a page, a view, or an object).

3.5.5.4¶4 1There are two variations, one has with screen diagrams and flows on the right, and a text description on the left. 2Charts accompany any items that require additional details. 3Every element needed is directly contained in the document. 4The other is much closer to the multi-page wireframe, with two or three images and annotation boxes with arrows pointing to the

The Weather Channel Mobile Web 3.0 High Level Design Document December 12, 2008 Version 121208 little springs design

## A Page Wrapper



The wrapper is the set of elements that surround the content on a set of pages. It is not the template, but items that are reusable between individual pages and page templates.

For this design, the page wrapper is essentially identical on all pages, including the home page for the site, and basically encompasses the header, footer and certain localization elements that launch from there.

Notable variations do exist between device classes, to which careful attention must be paid.

©2008 Little Springs Design - Proprietary & Confidential

### 3.5.5.4 ¶2 Arrange by sections to communicate the intent of each portion

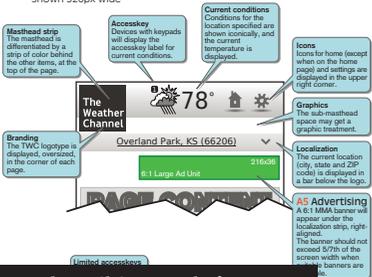
The Weather Channel Mobile Web 3.0 High Level Design Document December 12, 2008 Version 121208 little springs design

## A Page Wrapper > ii

### A1 Masthead

#### A1a Best-case devices

Shown 320px wide



- Masthead strip**: The masthead is differentiated by a strip of color behind the other items, at the top of the page.
- Accesskey**: Devices with keyboards will display the accesskey label for current conditions.
- Current conditions**: Conditions for the location specified are shown iconically, and the current temperature is displayed.
- Icons**: Icons for home (except when on the home page) and settings are displayed in the upper right corner.
- Graphics**: The sub-masthead graphic may get a graphic treatment.
- Localization**: The current location (city, state and ZIP code) is displayed in a bar below the logo.
- AS Advertising**: A 6:1 MMA banner will appear under the localization strip, right-aligned. The banner should not exceed 50% of the screen width when screen banners are limited.
- Branding**: The TWC logotype is displayed, oversized, in the corner of each page.
- Limited accessibility**: A note at the bottom of the diagram.

#### A1b Dynamic scripting

Shown 240px wide



- Smaller masthead**: The logotype, icons and strip are reduced in size, in steps, for classes of device screen sizes.
- No icons**: These functions are available in the footer instead (and usually via accesskey).
- Localization**: A pull-down selector menu is displayed below the masthead. The visible selection is the current location (city, state and ZIP code).

©2008 Little Springs Design - Proprietary & Confidential

### 3.5.5.4 ¶4 s4 Individual pages can have notes describing each feature...

E0735 USBN - Regulatory Changes to Registration & Authentication User Experience Design Update/Changes July 19, 2007 | version 4.07107

## 3 Registration Exceptions C> Off-Channel Notification Wait/Response

### Functional Description:

Whenever the web is used to send a reset code, this page is immediately returned to accept the code and allow creation of a new PIN.

A link to this page should also be included when possible (is in email) so that customers who expire the session may go there without sending another reset code.

The security question is also offered for updating on the basis that users who need an off-channel reset do not know or do not have a security question.

In the event the customer is not identified and would be presented this page, the personal greeting block would be removed.

### UI Details:

The communications method used (email in the example to the right) but not the address, will be dynamically populated in the introductory information.

The reset code is only good for use one time (once entered here it cannot be used again) and has an expiration time. There is, however, no limit to the number of times a code can be sent. When a new code is sent, the previous code is removed and can no longer be used.

The user must enter the reset code, and submit the (a) page. Errors on this page should be dealt with as a sign error, recycling the page with standard errors.

If the reset code is entered correctly, the (b) page is presented, and the user can view the string focus to the field) or create a new PIN and, if they wish, question and answer pair. There is no way around this page; clicking My Sprint or any other item will not get around it.

The fields within this page will operate as they do in all other locations; see 2A for a description of field behavior. The PIN will not be pre-populated, as it has been forgotten so it is to be discarded and replaced with the new one. The security question will be pre-selected, and the answer pre-populated (though marked when null) so the user may view the question without changing it if they so desire.




©1999-2007 Lucid Ent. | Proprietary & Confidential

### 3.5.5.4 ¶4 s1 ... or can be arranged with descriptions on the left, and drawings on the right

relevant portion of the image.

- 3.5.5.4¶5 1This document format was, in many ways, designed in the same way as any interactive product. 2One key finding after the first few trials was that the generalization of text-oriented vs. graphic-oriented people holds true. 3Text settled on the left side, or above the graphics, where the text-inclined may encounter it first, and consume it readily. 4Graphically-oriented readers will gravitate to the image, so it's okay to place it in other areas.
- 3.5.5.4¶6 1The annotation style works also works for all types of readers, but in a different way. 2The annotations serve as captions to the image. 3Those who are disinclined to decipher the image itself, can instead read the element title and the descriptive annotation boxes and retrieve much of the information still.
- 3.5.5.4¶7 1The text must be read by everyone, so generally starts with a basic, executive summary, then moves to extreme detail. 2Everything is written in specification format, with careful use of language to assure the meaning is clear.
- 3.5.5.4¶8 1Many portions of the process discussed — such as referring back to the high level document during creation — can help avoid the issues that might emerge from such a split up document format.
- 3.5.5.4¶9 1I also find that there is a design check and balance between the two sides of the page. 2Descriptions must match drawings, and vice versa. 3When revisions are made to the drawing, the text must be updated to reflect them. 4If you cannot explain it well, then perhaps it is not sensible, or is too confusing.
- 3.5.5.4¶10 1A revision, or alternative version of this includes the high level portions and style guide in a single document. 2The same organization is preserved, with a section introduction, then high level, followed by detailed, followed by style guide. 3While large, if the implementing teams work best like this, combine your documents for them.

### 3.5.5.5 Object Oriented Documentation

- 3.5.5.5¶1 1Object oriented designs actually don't work as well in the older, "ideal" modeling methods, like the annotated wireframe. 2Re-use of components has to be carefully explained or it will often be missed, and excess or inefficient development will result.
- 3.5.5.5¶2 1The specification styles of documentation work well for this, with distinct sections for each element, grouped by category, accompanied by detailed explanations of the use. 2The page wrapper (the header, footer and other components on each page) is a common first group to establish the detailed design.
- 3.5.5.5¶3 1Polymorphic components especially, with multiple variations, have to be specified as though they are tiny applications. 2Depict the underlying elements, and show the variations.
- 3.5.5.5¶4 1If you haven't designed from the front with reusable components, be sure to re-analyze the design after the first round of the high level design. 2Much like the development of a grid/template system during design, you can discover re-usable components and break them out from the individual page descriptions.

### 3.5.5.6 **Distribution**

- 3.5.5.6¶1 1I am going to avoid repeating too much here, but the distribution methods are key to making sure your design is implemented correctly.
- 3.5.5.6¶2 1In case you didn't read any other sections: version everything, distribute in a readable format (PDF, I always say) and post somewhere that anyone on the team can get it (avoid email, use password-protected websites).
- 3.5.5.6¶3 1Delivery is an active condition; do not just post files, but have walkthroughs of the design, and be available for questions and revisions. 2There are numerous additional guidelines on this later.

## 3.6 Style Guides & Graphic Deliverables

3.6¶1 1In the past, any time we went far enough to create production graphics, they were simply packed up and emailed, then often ignored or misunderstood. 2Anything else, like colors or type sizes that needed to be complied with, got included in the detailed interface design document, where they were generally lost in the shuffle.

3.6¶2 1I usually worked around this by conducting clear walkthroughs, and arguing that anything specified has to be built, so opened bugs on design errors. 2But this is clearly the wrong thing to do. 3So lately I have been regularly building style guides instead.

### 3.6.1 Style Guides

3.6.1¶1 1Prototypes, even of absolutely full fidelity, are not useful to define styles for implementation teams. They will have to devise some method to measure space, size and color.

3.6.1¶2 1Instead, what you need to complete the specification enough to get the product implemented is a style guide. 2This term means nothing to a lot of people. 3I have never investigated why and have no idea what else to call it, so end up just explaining it. 4Since the rest of your deliverables will also need to be explained during the sales or scheduling phase, this is not a big impediment.

3.6.1¶3 1The style guide is very much like a brand document. 2It will have sections on each aspect of the visual language used to create a cohesive design. 3Type, sizes, colors, margins, spaces and everything needs to be outlined.

3.6.1¶4 1Style guides may also need sections on content creation, including written content. 2There may need to be guidelines for length, compression, and resolution or contrast.

3.6.1¶5 1Your style guide must exist within the brand, and the product set within which the project resides. 2For some projects, you might have to develop much of this from scratch, or work with other teams to create brand guidelines. 3Other-

**3 Masthead > Logotype staging and sizing**

The most obvious element to communicate the brand of the Flora service to any users is the logotype in the masthead (and footer) of every page. The specifications for the use of the logo in these two locations are discussed in the right.

**Two sizes** – The logo will not scale depending on the device, or for any other purpose. To account for different device sizes, there are two logotypes available. These are detailed in the other footer.

**No background** – The masthead has no background and all content is centered. The logotype is simply on a white, default background.

**Other page elements**, such as the H1 title immediately below this, will offer additional branding and opportunities to place the brand colors.

**Includes in markup** – The logo should be included in markup as an image element. This will ensure the logo is always visible, regardless of style, support for backgrounds, loading of external style sheets and so forth.

**Not a link** – Some devices will force a border around every image that is a link, regardless of styling. To avoid this happening on the branding, do not use the logo as a link.

**Staging area** – The area around a graphic that is specified for its position on the page is the staging area. The staging area is the area that is centered, and stretched by several elements. The left and top margins are set by the page layout (see section 2, above). The bottom and right margins are defined by the other elements.

Some text elements can occupy the space to the right of the masthead on some pages. The right margin controls how close these elements can approach the masthead, after this they will overflow by wrapping to the next line.

**Footer** – Though not shown, the footer logotype will be the same size, with the same staging area and other requirements.

	Screen width	Logo size	Bottom	Right
Small	176 or smaller	80 x 20	7 px	5 px
Large	177 or larger	110 x 30	10 px	15 px

©2008 Little Springs Design - Proprietary & Confidential

**4 Typography > Fonts, faces and sizes**

Typography on mobile devices is mostly frustrating and tightly constrained. While some devices – like iPhone – offer a variety of scaled faces similar to that available on the desktop, most have only a single face, with limited styling. Design and implementation should always take into account poor control over type, and allow for unpredictable spacing, wrapping and other overflow methods.

The guidelines outlined here presume two different display size ranges, and cover both limited devices and those with more type options. A chart is provided for comparing the three basic size systems.

**Typeface** – As with desktop websites, specialized fonts like DIN cannot be used, as they are not loaded onto the client. Since most devices have only one font within the device, there is often no point in adding font family.

**Use Arial or Helvetica** – When type can be specified (e.g. iPhone) use Helvetica, then Arial. These are common, readable faces that are within the DIN family, so reflect the Chicago branding. See examples in the screenshots to the left.

**Additional styling** – Most devices will only support bold and normal weights, and no other styles. Even Helvetica is not universally supported. For this reason, the styling specifications do not reflect any type treatments but size and weight.

If the design can be enhanced by the use of special styles, that is acceptable but it should be recognized that they will not work on most devices.

**Use relative sizes** – For the most part, type should be specified in relative sizes. That family names, and scaling (such as sans preferences for type) will be system dependent.

**Test Staging** – Most mobile devices support only three type sizes: small, medium and large. The default design document uses only these sizes (small, medium and large), and this should be carried through to all instances, regardless of which device is used to test variations in the device.

For compatibility with devices that support specifying other type sizes, the chart below is provided in all of the pages, when needed that correspond to the three basic sizes.

Also note that for the three sizes, often different screen resolution systems are available. The resolution dimension is used to determine screen resolution here.

©2008 Little Springs Design - Proprietary & Confidential

	Normal	Extra	177 px and wider	Pixel
Small	14 Helvetica bold Italic	16 Helvetica bold Italic	14 px	12 px
Medium	16 Helvetica bold Italic	18 Helvetica bold Italic	16 px	14 px
Large	18 Helvetica bold Italic	20 Helvetica bold Italic	18 px	16 px

**6 Color > Brand compliance and legibility**

A key element of branding is utilizing brand-associated colors, and supporting colors and colored graphics, with all aspects of the service.

This must – especially for interactive systems – be balanced with the needs of legibility. Text must be readable without undue effort, graphic items must not be lost in design, and the manner of interaction must be clearly understood.

For this reason, the design has a white background with black text. Green is used for brand accents – as the title bar background, for bullets and so on – and brand-supporting blues are used for links, when available.

The chart below depicts the approximate percentage distribution of color on a typical page, assuming color fills and discounting raster images.

Flora logo green	# 153	Hexadecimal	#99CC33
Flora light blue	# 204	Hexadecimal	#9999CC
Flora dark blue	# 153	Hexadecimal	#003399
Black	# 0	Hexadecimal	#000000
White	# 153	Hexadecimal	#FFFFFF
Grey	# 153	Hexadecimal	#999999
Flora orange	# 255	Hexadecimal	#FF9933

©2008 Little Springs Design - Proprietary & Confidential

**7 Page Elements > Styling for everything else**

Individual page elements are all designed and coded to function without styling, to assure functionality on all devices. This will also encourage better access by search engine bots, screen readers and for unpredictable situations, such as poor connections.

Elements used within this design may be broken into three categories, based on the desired modification from defaults.

**Unmodified elements**

**Form elements** – Items like checkboxes, radio buttons, pull-downs and form fields will be used unmodified. These forced styles may range from default styles and conforming to user-defined styles, such as poor connections.

**Bullets** – Bullets are used to apply styling to bulleted points. The method of styling bullets – using image markers or the image '•' in text, therefore, to accept the default display. Bullets can, however, effectively communicate additional information (e.g. primary vs. secondary) by attaching characters to the bullet field, such as the detailed design, and the chart below for entries to place before the label text.

Here are examples:

Label	name	label	text
add	add	add	add
cancel/remove	cancel	cancel	cancel
okay	okay	okay	okay
continue	ok	ok	ok
back	back	back	back

**Sometimes styled**

**Links** – Links are very often styled by the mobile browser, and any design-in style is damaged. These forced styles may range from default styles and conforming to user-defined styles, such as poor connections.

For devices that will style by design, styling, unmodified links with underline, and the attributes for underline:

- 1 An unvisited link
- 2 A visited link
- 3 A hovered link
- 4 An active link

**Bullet lists** – Bullet lists will, on some mobile browsers, revert to standard display. Most will comply with the suggested styling, which focuses on removing unwanted space. The list margin, as well as additional space above and below the list, should be removed.

The default bullet will be replaced with Thick dark blue (#003399) (online: 003399; hex: #003399).

1.25 em of the previous item.

0 List items wrap to hang on the bullet.

0 The last list item end of the previous item.

1.25 em of the previous item.

0 vertical padding

0 margin

0 Save changes

0 Cancel

©2008 Little Springs Design - Proprietary & Confidential

3.6.1 A style guide showing logo grid, type, color and other details often missed in technical specifications

wise make sure you get ahold of all these other documents, not just before you write your style guide, but before you start the project.

3.6.1

1 Remember, these are guides and guidelines. You should have enough leeway within these other styles to create your design; but you also need to create your style guide to give developers, content creators and unforeseen future events the flexibility they might need.

3.6.2

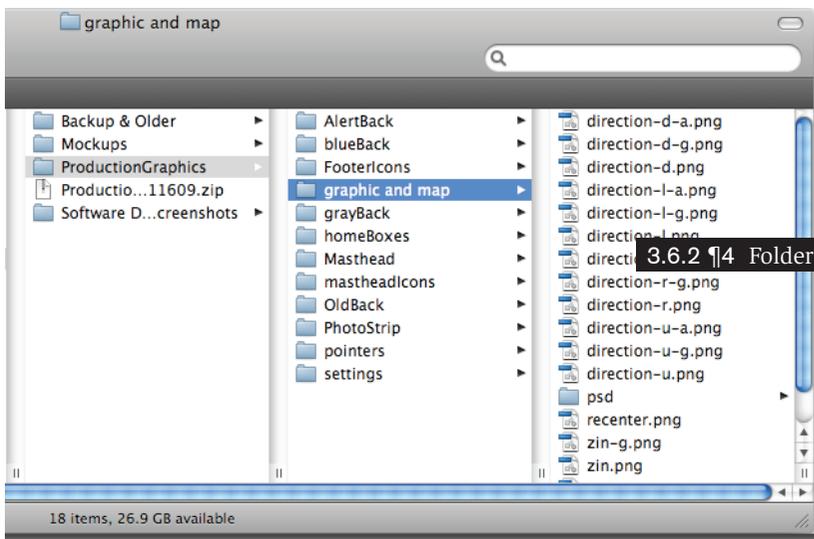
Production Graphics

3.6.2.1

1 Pretty much all I have to say about sharing graphics is the same as sharing prototypes, or other design documents. 2 Make it available to everyone, keep it versioned, etc.

3.6.2.2

1 There are a few more complexities, as there



will be many files. **2**It's usually best to deliver all graphics as a single compressed package, for speed and to encourage everyone to get the full set of files.

**3.6.2¶3** **1**But once that is opened, try to help the developers or downstream designers by organizing and labeling well. **2**In the example, note that I have divided the graphics into folders, for starters. **3**These reflect the organization, and the name

- 3.6.2¶4 1Similarly, name your graphic files in a useful manner as well. 2Try to keep them short, but label the basic item and follow up with codes for direction, location, background color, size or whatever variations may be important.
- 3.6.2¶5 1I also like to include the editable base files. 2Usually this means Photoshop. Even if the client doesn't have the capability to edit, or the skills, it shows off how hard you are working and demonstrates a trust and open-ness on your part. 3More often than not, doing this will cause change requests to come back to you, but at least there is a chance to get changes that look good, instead of risking some marketing guy's daughter up-scaling exported images, poorly (this has happened).
- 3.6.2¶6 1The style guide should be the legend, and instructions for all of these graphics. If there is no style guide, include that information in the detailed design document. 2Check, before you send it off, that graphics are named consistently with the document, and important items needed to use them are included. 3If there is a strong chance that a downstream team will use your Photoshop documents, make sure information on how to export is also included.

## 3.7 Prototypes and Presentation Layer Deliverables

3.7¶1 1Prototyping can be a valuable exercise, at the right time, and for test purposes. 2Some design teams will produce the final code for the presentation layer. 3This usually joins to the graphic deliverables outlined above, but they are separate exercises, however intertwined they may become.

3.7¶2 1Above all, remember that all of this is still design, and design specification, work. 2The prototype and the code are an end product only in a very narrow view. 3Their reason for existence is to support the design of the final, functional product.

### 3.7.1 Prototypes

3.7.1¶1 1Prototypes are never complete representations of the system, and are therefore just sketches of the final product. 2They must, most of all, clearly be not production code. 3While ideally they run on the same platform as the final product, or a reasonable facsimile thereof, try to specifically avoid using the agreed-upon technology. 4Often, this is easy as Flash and PHP are easier to develop in than C++ and Java.

3.7.1¶2 1Present prototypes as you would any other design document. 2Walk the client through it, demonstrating the salient features. 3Never demonstrate the features of the prototype, but of the design itself. 4Be sure to mention when there is a gap in the prototype, or some limit on information or behavior, so they can imagine how the real product will be extended. 5Developers especially will often take a prototype excessively to heart, and not code for the designer's intended dynamicism.

3.7.1¶3 1Some clients will require specific types of delivery mechanisms, but I generally put all my deliverables on a password-protected web server. 2A short email can be passed around the company with the link and credentials. 3If the deliverable is tricky to install, or the hardware is not common (or is restricted) as for many mobile devices or appliances, you will probably have

to install it onto a limited number of devices yourself. **3**These are usually best shared by setting them up in a specific location in the client site. **4**Sometimes they need to be passed around, but either way give them a home, with supporting information, and other documentation. **5**You can even make a sort of science-fair kiosk with a table and posters to emphasize what the prototype is and is not, and remind everyone to look at it.

### 3.7.2 Code

**3.7.2¶1** **1**One team I ran, and several I have worked with, produce presentation code as a separate exercise from the development of the software. **2**For websites, this means all the html, css, javascript and Flash. **3**Graphics are discussed in the style guides section above, but generally also have to be created to make sure all of this code works.

**3.7.2¶2** **1**The differentiation between presentation layer and other software and data layers is a useful one for semantic purposes and for purity of software and data. **2**It will help keep bugs from tangling with other parts of the system, free each layer from the others for fixes and upgrades, and so on.

**3.7.2¶3** **1**It is also useful in that the presentation layer can be built as flat and stupid, and/or attached to stub data sources and become a sort of complete prototype. **2**This should be installed somewhere where the entire project team can view it and use it, as with prototypes. **3**If this code is released before it is completely final, or if any serious changes are made during development, provide for a versioning system. **4**For anything served over the internet, require access through a page that lists all of the available versions so the developer can see when changes are made.

**3.7.2¶4** **1**However, the deliverable to the technical implementation teams will be something else. **2**You have to talk with them about their process, but generally it will be flattened scrapes of the clever code your team creates. **3**Since you presumably used good modular, object-oriented practice, avoid letting them use — for an example on the

web — each page as a complete element. <sup>4</sup>Try to make them understand, and try to deliver, componentized objects.

- 3.7.2¶5 <sup>1</sup>The delivery of coded components should tie to the other design documentation; often, these components can follow the same numbering scheme (e.g. *A3a.inc.html*). <sup>2</sup>Actually, using code like this will require the implementation teams to go find the decrypt table, and review the design documentation.

## 3.8 Delivering and Presenting Artifacts

3.8¶1 1First, don't do it alone. Ideally, get a project manager, or a team of people, to help. 2Even if you are good at client relations, it takes a lot of time. 3For project meetings, at the very least borrow someone else to take notes while you do the presentation itself.

### 3.8.1 Labeling

3.8.1¶1 1Most of your artifacts will probably be of the multi-page style, and as mentioned before should follow a presentation style. 2This means good organization, clear statement of the scope and purpose of each page and section, and clear labeling.

3.8.1¶2 1Most labeling, such as numbering and naming elements, has been covered already. 2But also pay attention to the title page, headers and footers.

3.8.1¶3 1The title page should communicate your team, the client, the project and any other important information that will help to clarify for the project teams, such as the domain. 2Avoid too much graphical flash, and make sure that it's all readable small, when photocopied and so on. 3Make sure version numbers and dates are visible.

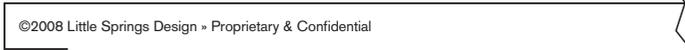
3.8.1¶4 1I am indecisive on whether team member names should be placed anywhere in the document. 2For design teams that are integral to a large company, this can be useful because it may not be clear who to contact otherwise. 3There can be other, political uses as well. 4One time I had to present a design to some skeptical marketing research types. 5Though I did most of the drawing, I included the names of the rest of the project team, and made a point of including the "PhD" title for the Human Factors representative. 6Normally I do not put names on, as the team stands behind the deliverables as a whole.

3.8.1¶5 1Each page (aside from the title page), should have headers and/or footers that communicate in smaller type the same basic information as the title page. 2In traditional wireframe sensibil-



ity, many people I know insist these should be black and white also, but I think it's okay to express the brand of your company (whether internal or an agency) here. **3**At Little Springs, we even include our logo, though very small.

- 3.8.1¶6** **1**The footer of each page should also include copyright notices, and a generic line about the information being proprietary and confidential. **2**It won't stop anyone malicious, but gives you some cover and reminds everyone that the pretty pictures are delightfully concise summaries of some of the the problems and system architectures of your company, so don't leave them in taxicabs.



- 3.8.1¶7** **1**At agencies (vs. most internal organizations) it is also a good idea to have a page at the end that summarizes who you are, and gives your contact information. **2**A lot of people start presentations and deliverables with this information, but I think it's annoying to clients. **3**If anyone wants the information generally, or finds your document in future years and wonders how to contact you, they will find it at the end.

**3.8.2¶1**  
 The key to presentations is to remember that you are presenting; even when showing off a specification, your talk is more important than anything else. If technology fails you, then just keep going. For a lot of really good tips, visit the blog (and get the book by the same folks) *Presentation Zen*.

- 3.8.1¶8** **1**I also think a concluding page is good for any presentation. **2**Since the organization of the diagrams and specifications will tend to put the most boring information last, it's hard to end on an up note, so at least the team information page serves as an adequate "this is the end" slide.

**3.8.2** **Presentation Tips**

- 3.8.2¶1** **1**Presentations should always be in person. **2**If not, then at least on the phone, and try to set up some sort of internet meeting, so you can drive the pace of the presentation and point at interesting features. **3**Worst case is that you simply talk and refer to the frame numbers to guide everyone through the document.

- 3.8.2¶2 1Try to get everyone in the same meeting, but if you need to, have separate meetings for implementation and marketing and other teams. 2The risk of this is that I.T. wants changes that you then need to take back to the business owners. 3Make sure the important people get the presentation. 4This is not always, or even usually, the most senior people. 5Find out who will be doing the actual work, and try to meet with them.
- 3.8.2¶3 1But do have meetings regardless. 2If you just send the documentation out, it will be poorly read, and even more poorly understood. 3Always be happy if someone finds errors or inconsistencies. 4It means people are reading your document in detail. 5They are your friends, as they are likely to actually follow the specification and implement what you have asked for.
- 3.8.2¶4 1The presentation should be like a presentation. 2Consider the audience and point up the parts they care the most about. 3For a project team meeting, often there are lots of different people, but if there is someone much more senior, appeal to that person; everyone will expect it. 4Keep the meeting on pace, so you cover the important parts in the time allotted. 5If you get done early, all the better; there's time for questions, and even if not, it makes it look like your document is simple and easy to understand.
- 3.8.2¶5 1There's no way to walk through the whole document in most cases. 2That will take several hours, and bore everyone. 3But still use the real document instead of any slideshow. 4You need to emphasize that the document is the deliverable, and there are no shortcuts.

### 3.8.3 Procedures

- 3.8.3¶1 1I always, always export files as PDF. 2Even if your company loves Visio, someday someone won't have it, or will need to look at it on their mobile phone, or their home computer, or their version will fail. 3But everyone can look at PDFs. 4The viewer also has a pretty good slideshow mode.
- 3.8.3¶2 1If you do this a lot, get Acrobat Pro. 2It lets you compress files more, pick a version to export into

(pick a low version, to make sure everyone can read it) and so on.

- 3.8.3¶3 1Ask for feedback. 2Especially when people don't know you, and sometimes if your presentation is so good you look like everything is perfect, no one will ask a question. 3Try to have questions of your own to ask the rest of the team.
- 3.8.3¶4 1Make sure you ask if the technical teams, and brand and everyone else on the project team has seen it, and approved it. 2Ideally, get signed approval, but since this never happens, try to get explicit verbal approval, and write it down for later.
- 3.8.3¶5 1Your organization will have procedures to follow schedules, so ask your project managers about defining the end of a phase, and client acceptance. 2They are better at this stuff, but someone needs to do it, and if it's you that's better than no one doing it.
- 3.8.3¶6 1Follow up meetings with emails, outlining that a delivery was made (this puts it down in the record) and with what was discussed, who was there, and any open issues. 2Save your outbound emails; this doesn't happen automatically for the most part, so pay attention to this stuff.

### 3.9 Iterating on Detailed Design

- 3.9¶1 1Your design better not take place in a vacuum. 2While this process helps you design on your own, and has many methods to check and improve independently, you should bounce them off others, and almost certainly have to get buy in from product owners, and get technical teams to build the product.

#### 3.9.1 Iterating on Design

- 3.9.1¶1 1Iteration is revisiting design to improve, or add to the completeness of the total product. 2Some simple suggestions I have for iterating on a design are:

- 1 Set it aside. Try to wait at least a day.
- 2 Show it to someone else
- 3 Do it again. And again, and again. Iterate.

- 3.9.1¶2 1Try to do all of these for each design document, before delivering it and before showing it off. 2Feel free to play, and make two or twenty high level designs in one day, before you show it to anyone. 3And do show it to people. 4When possible, collaborate with other designers, use their expertise if you know they are better at a certain aspect of design or systems.

- 3.9.1¶3 1Don't just accept or reject any designs you come up with, or any element of them. 2Know why you are doing what you are doing. 3Go back to the top of the process, and look at the morphology chart, look at the design principles. 4Consider how any solution meets, or violates the basic principles, the design concept, the current flow chart.

- 3.9.1¶4 1Don't just increment, and come back to add features, but iterate by considering the entire design, and what adding or modifying a feature or design element does to the whole.

- 3.9.1¶5 1Don't be afraid of failure. 2Do try to avoid the project failing, but within your conceiving, brainstorming, and sketching, try out new ideas. 3Frequently, I will throw up a stupid idea a dozen times on a dozen projects before I figure out what it is that caused it to be stupid, and then

realize it's a perfect solution for this project. **4**A fear of failure would eliminate that experimentation.

- 3.9.1¶6** **1**If the design is not correct immediately, and it probably won't be, then go ahead and do it again. **2**Try to evaluate why it didn't work so you learn, but mostly be willing to sketch, draw and design more.

## **3.9.2 The Project Lifecycle**

- 3.9.2¶1** **1**The other key method of iterating, or collaborating is that you are (probably) not a mad scientist or loner artist. **2**There is, as mentioned many times above, a project team that has to believe in the design. **3**You also have to believe in their needs, their concepts, their ideas and constraints.
- 3.9.2¶2** **1**And you have to work within the timelines set forth, and the progression of the project as it proceeds through it's development, construction and launch.
- 3.9.2¶3** **1**To do this, you have to communicate with the rest of the team. **2**Much of this is suggested by the above processes, the rest is required. **3**You have to gather information from the business stakeholders. **4**You really should work with development and test and other technical teams to design systems they can actually build, and that will work efficiently.
- 3.9.2¶4** **1**Be open, and flexible. Insisting that a design detail cannot be changed will simply get a larger set of features cut, or the developers will come up with their own solution. **2**Usually a terrible one. **3**Work with them. **4**Nurture everyone's strengths. **5**When problems come up, seek the core issue; technical issues are usually not what is stated. **6**More often than not, I have been able to explain how the design specification was simply written in such a way that it worried the developers, without being impossible to build. In the end, no change required.
- 3.9.2¶5** **1**In all cases of change, remember the ideas outlined just above about going back to basics. **2**Never just solve a problem off the cuff. **3**Seek alternatives, and prepare for solutions to issues

you think might come up. 4If an issue is not settled entirely, make an alternative design you can pull out if they again threaten to descope or change it, even months later.

3.9.2¶6 1Be available for consultation, and respond quickly. 2Understand their schedules. 3Not only do you have to respond quickly near the end of the development process, but often there are strange work schedules, such as overnight test and daytime development; if someone on the design side can make it clear they will pick up the phone for the test team at night, you have a better chance of getting the product made the right way.