

Analytic Synthesis and FLiP: How to Eat an Elephant

Jeff Peters

jeff@GrokFusebox.com



How Do You Eat an Elephant?

One bite at a time.

but...



How do you *bite* an elephant?

When the solution to a problem is not clear, we need a way to get from the whole problem to solvable pieces, and then to increasingly complete solutions.



Analytic Synthesis (AS)

- ♦ **Everything you do consists of analysis, synthesis, or both.**
- ♦ **AS is the discipline of becoming aware of the fact that you do this.**

Analysis

- ◆ "Break it down"
- ◆ Ask questions
- ◆ Challenge assumptions
- ◆ Understand the problem



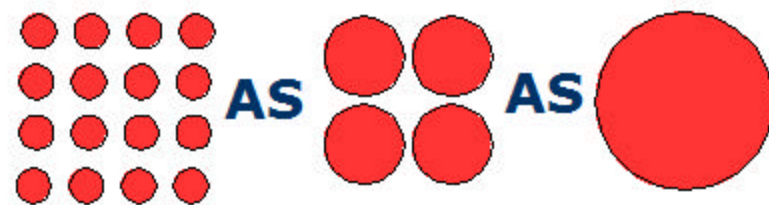
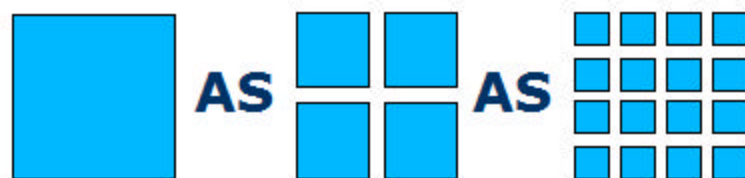
Synthesis

- ◆ "Build it up"
- ◆ Answer questions
- ◆ Validate assumptions
- ◆ Understand the solution



AS is Iterative

- ◆ **Bigger to smaller**
- ◆ **Smaller and smaller...**
- ◆ **to smallest**
- ◆ **- and -**
- ◆ **Smallest to bigger**
- ◆ **Bigger and bigger...**
- ◆ **to Done**



Best Processes Exhibit AS

- ♦ **Formalized processes that recommend analytic steps in preparation for synthetic steps work naturally toward a goal.**
- ♦ **- and -**
- ♦ **Formalized processes that attempt synthesis without analysis often fail without indicating why.**

So What?

- ♦ **The concepts in AS are not revolutionary,**
- ♦ **- BUT -**
- ♦ **Understanding when *you* need analysis vs. synthesis is.**

FLiP Reflects AS

- ◆ **F**usebox **L**ifecycle **P**rocess
- ◆ **B**reaks "Build a Web App" Down
- ◆ **D**efines analytic and synthetic steps
- ◆ **I**terates as appropriate



FLiP Process

- ◆ **Wireframe**
- ◆ **Design Templates**
- ◆ **Storyboards**
- ◆ **Prototype**
- ◆ **Architectural Design**
- ◆ **Fusedocs**
- ◆ **Coding**
- ◆ **Unit Testing**
- ◆ **Integration/Deployment**



Wireframe

- ◆ Analyze client demands
- ◆ Synthesize process

WireFrame Viewer/Editor 3.70

"wireframe"

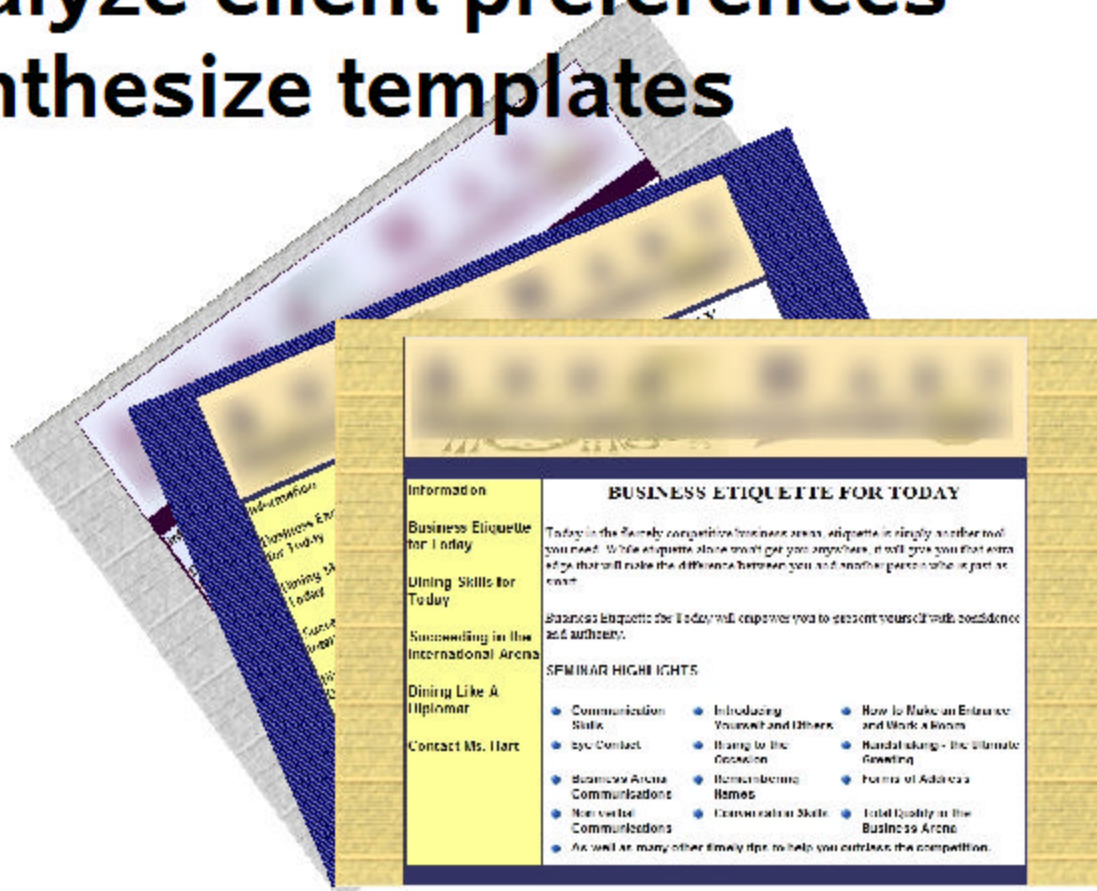
File Edit History Generate Docs Help Credits

Quick Map of "wireframe"

Page	Exits and Destinations
Home I show a quick map of the Application in the form of a list of ALL pages/states, together with their descriptions. The pages are also hyperlinked to the wireframe for the next page. In this way, the user can quickly jump to any point in the layout. FORM: I also show the entire wireframe details in a form. There is also a field to indicate a short text description of any changes made. The entries can be edited, and the new version is stored.	Jump to a page/state ----- Quit ----- Guid: Map History Help
Show PageState I show the responsibilities of a particular page/state. I also show a list of links from this page/state, in the role of content. The links can be generated.	Follow a LINK to ----- Quit: Map

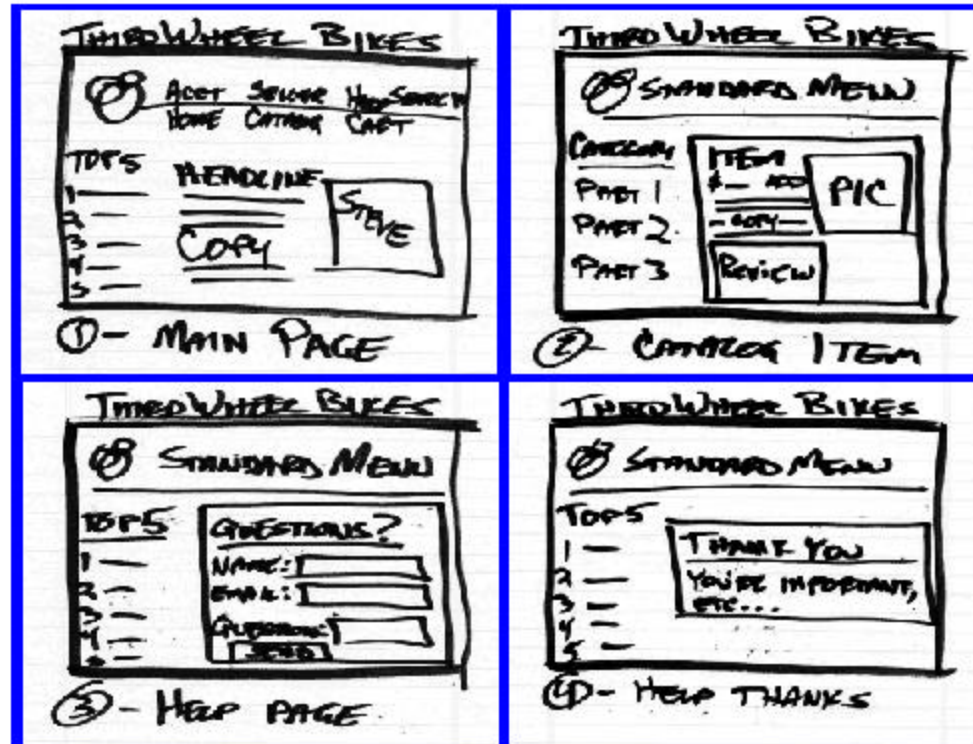
Design Templates

- ◆ Analyze client preferences
- ◆ Synthesize templates



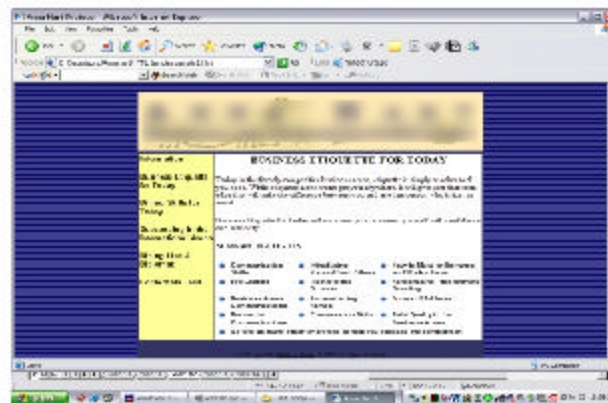
Storyboards

- ◆ Analyze wireframe
- ◆ Synthesize pages



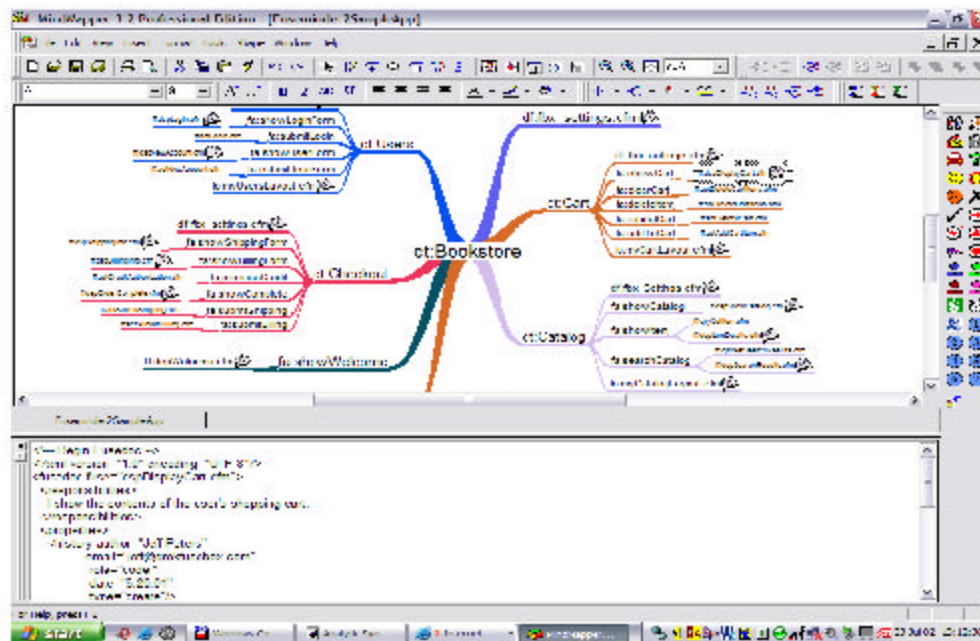
Prototype

- ◆ Analyze design templates and storyboards
- ◆ Synthesize prototype



Architectural Design

- ◆ Analyze prototype
- ◆ Synthesize architectural model



Fusedocs

- ◆ **Analyze architectural model**
- ◆ **Synthesize Fusedocs**
 - ◆ Instructions to developers
 - ◆ "Blueprints" for code

Unit Testing

- ◆ **Analyze Fusedoc**
- ◆ **Synthesize test harness**
- ◆ **Analyze test results**
- ◆ **Synthesize corrections**

Integration Testing

- ◆ **Analyze test results**
- ◆ **Synthesize bug fixes**

qsBlog – A Small Example

qsBlog is a Fusebox circuit that allows you to maintain a web log ("blog"). It was written to demonstrate the use of Hal Helms' QuerySim tag to manage data, hence the "qs" of qsBlog.

FLiP through qsBlog

