

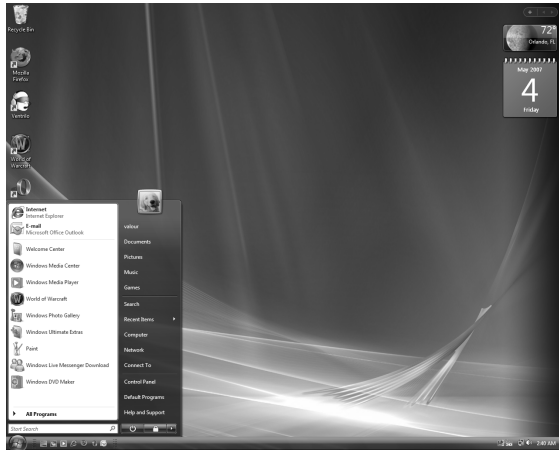
Rapid Ideation and Creative Problem Solving
Designing Software

Matt Franks & Lauren Serota



ac4d

What makes software design unique?



Microsoft Windows Vista



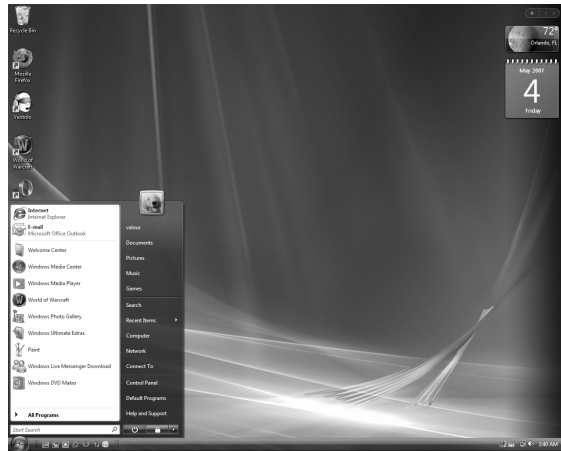
Black and Decker Toast-It-All Plus

How are these things similar, related to making them?

How are they different?

Why are they different?

What makes software design unique?



Microsoft Windows Vista



Black and Decker Toast-It-All Plus

How much does it cost to add or change features?

When can features be added or changed?

How many people are required to make this?

How many things can you do with this?

How can you tell if this was made well?

How much does it cost to add or change features?

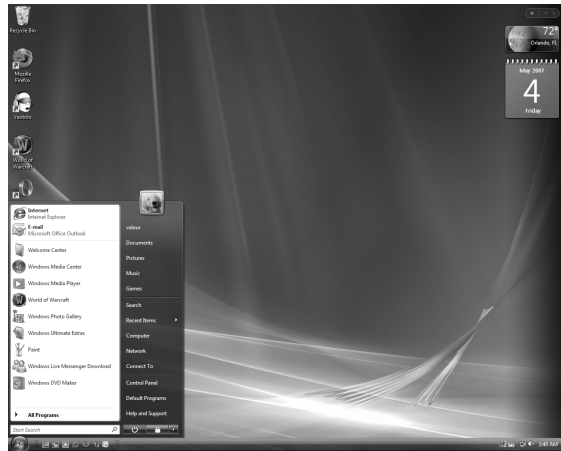
When can features be added or changed?

How many people are required to make this?

How many things can you do with this?

How can you tell if this was made well?

What makes software design unique?



Microsoft Windows Vista

Fragile	Complicated	Infinite
Layered	Customizable	Intangible
Unique.?	Hidden	Surreal



Black and Decker Toast-It-All Plus

Firm	Simple	Finite
Flat	Contained	Physical
Copied	Obvious	Real

“Design disciplines are concerned with making artifacts for human use ... One of the main reasons most computer software is so abysmal is that it’s not designed at all, but merely engineered” (Mitch Kapor, 1990).

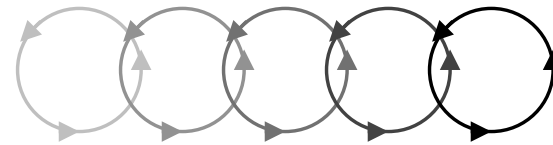
Software Design Methodologies

Although there are many variations on these, there exist two main schools of thought about how Software should be designed:



Waterfall Software Development

- Few iterations
- Spend as much time planning as possible
- Design, the stylize, and then code
- Maximize thinking, and minimize doing
- Emphasize documentation



Agile Software Development

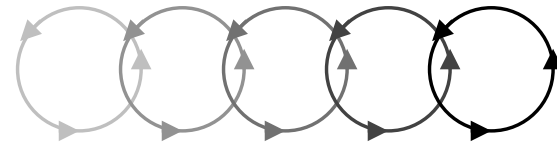
- Many short iterations
- Create something real as quickly as possible
- Code, design and stylize at once
- Spend less time planning and more time doing
- Emphasize face to face communications

What are the pros and cons of Agile?
What are the pros and cons of Waterfall?

Which do you think is better?

Software Design Methodologies

An Emerging hybrid process looks like this:



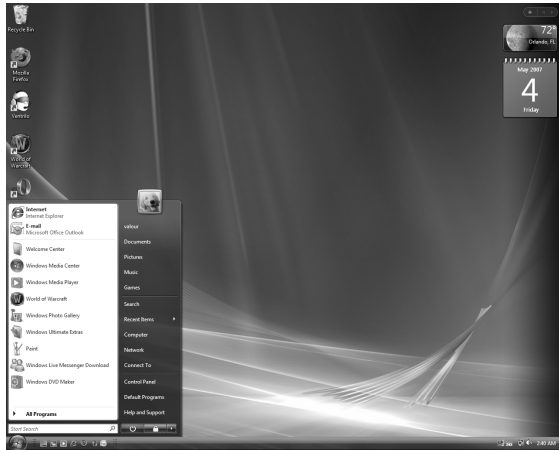
Research and Synthesis

Design and Development

What are the pros and cons of this hybrid process?

How do you think it compares?

What makes software design unique?



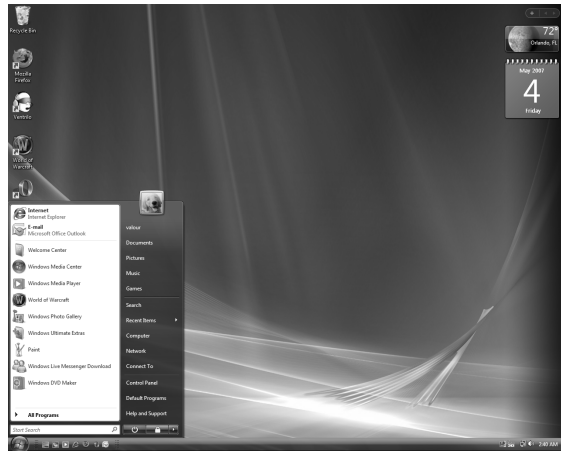
Microsoft Windows Vista



Black and Decker Toast-It-All Plus

What are some examples of artifacts generated during the design, development, and production of these products?

What makes software design unique?



Microsoft Windows Vista



Black and Decker Toast-It-All Plus

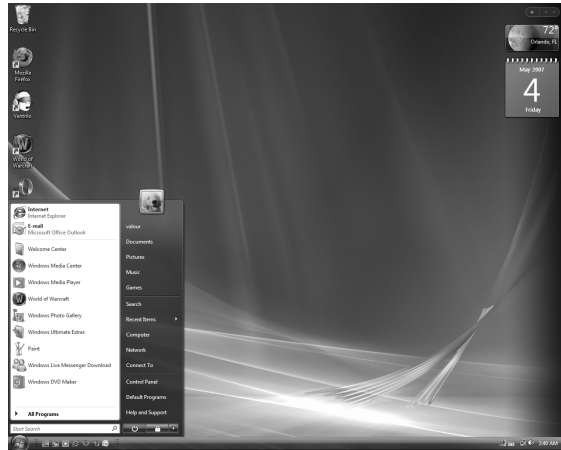
What are some examples of artifacts generated during the design, development, and production of these products?

Sketches	Concept Models	Interaction Models
Wireframes	Prototypes	Code

Sketches	Digital Models	Physical Models
Mechanical Drawings	Schematics	Tools & Dies

What are all of these things, and why are they useful?

What makes software design unique?



Microsoft Windows Vista



Black and Decker Toast-It-All Plus

What is a Concept Model, why do we use it, when?

Sketches	Concept Models	Interaction Models
Wireframes	Prototypes	Code

Sketches	Digital Models	Physical Models
Mechanical Drawings	Schematics	Tools & Dies

What is an Interaction Model, why do we use it, when?

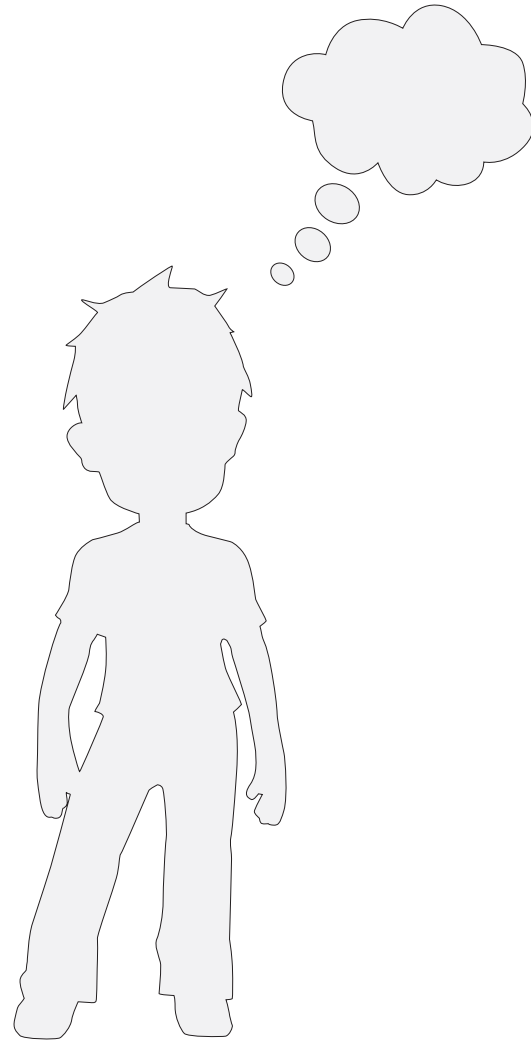
What makes software design unique?

A Concept Model

Is a diagram used to depict the users mental understanding of the system.
(aka their mental model)

Why would we create this?

When is it useful?



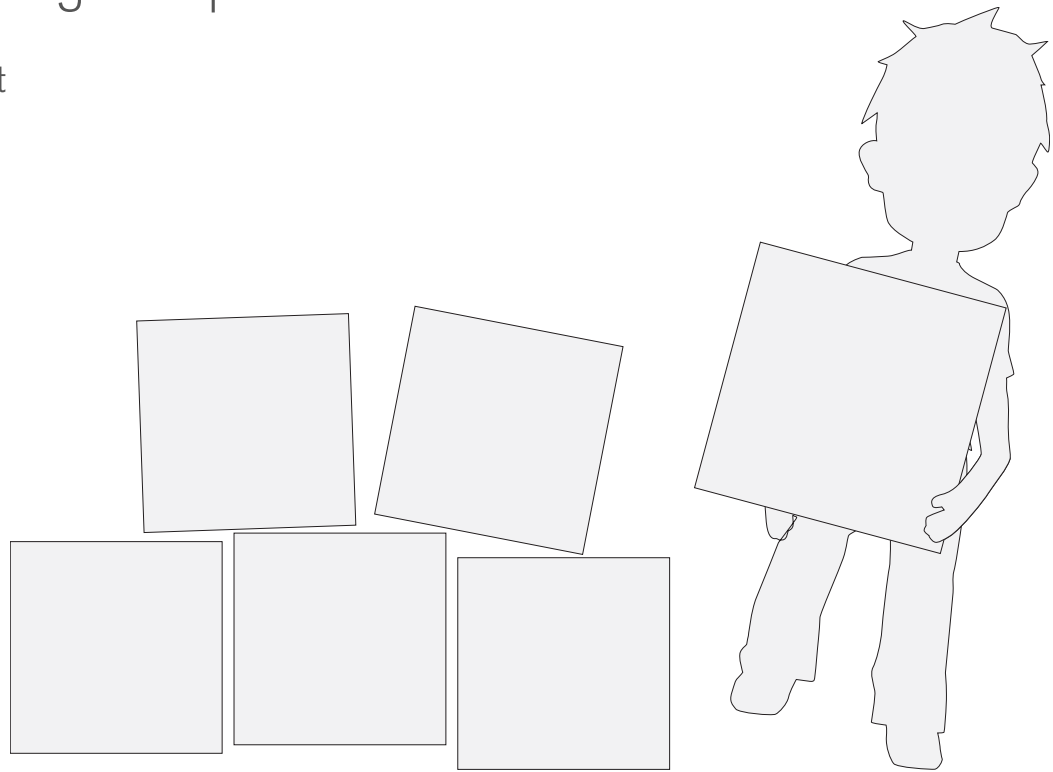
What makes software design unique?

An Interaction Model

Is an expression of the concept model through architecture, hierarchy, and navigation patterns.

(it defines the rules of the system without doing it within the context of wireframes)

Does 'expression' mean that there is more than one version?



What makes software design unique?

Interaction models are generally used when doing really conceptual work. Why would this be the norm?

- It depicts the core components of the system
- Clearly articulates the relationship between components and the larger system as a whole
- Has a clear set of navigation patterns / follows existing navigation paradigms
- Leaves room for interpretation - A variety of screens can be designed within the same set of rules and relationships

Why would we create an interaction model?

When is it useful?

If you want it done right, do it
yourself

Should we all learn to code?

Should we all learn to do everything?

ac4d