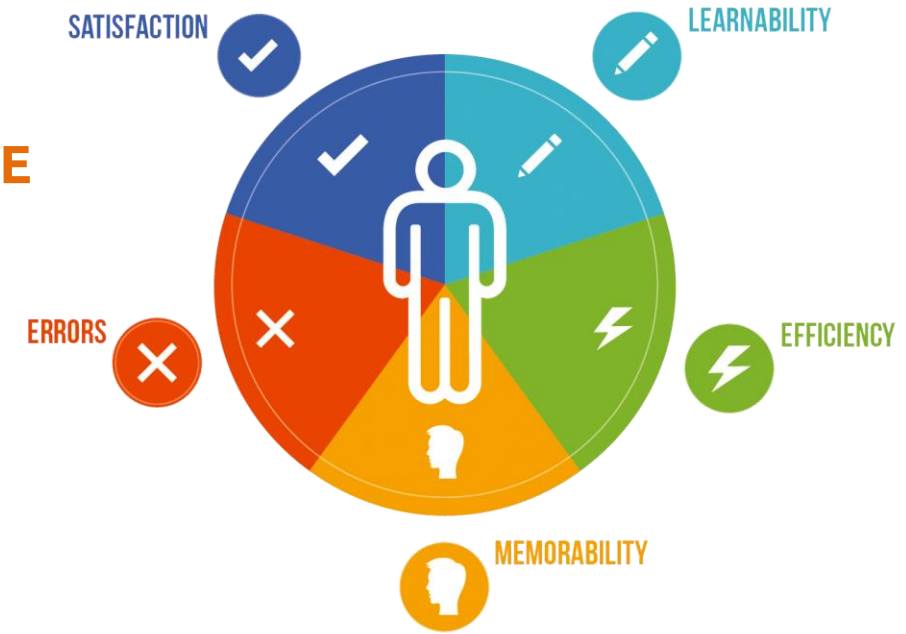
User-centered design

DESIGN FOR USER EXPERIENCE



User-centered design

- Definition
- Human Computer Interaction
- User experience
- User-centered design process for web sites

User-centered design

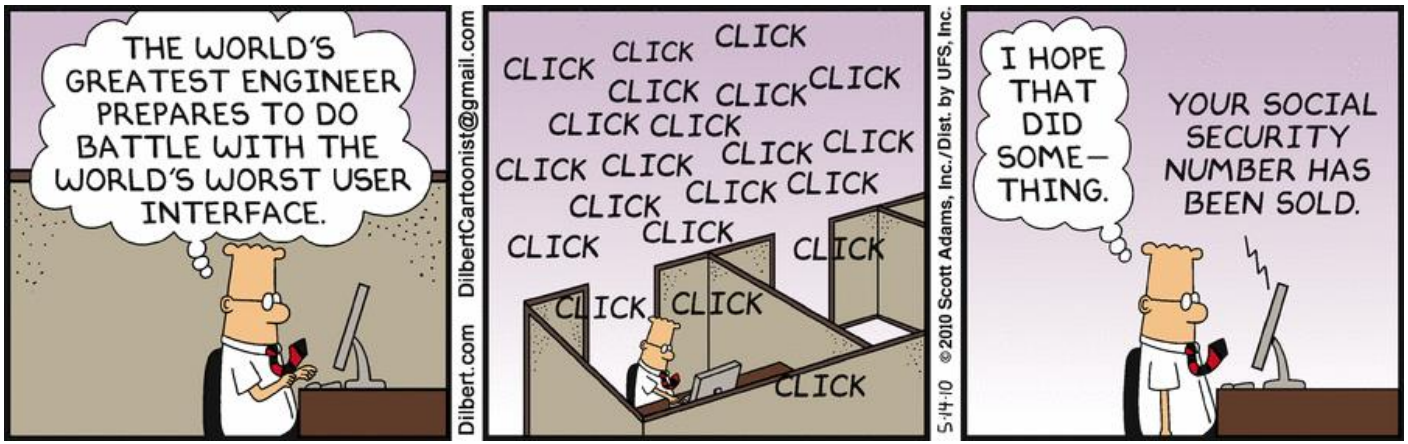
- A philosophy of product development
- The product is not an end in itself: it is a means toward the end of providing a good experience for the user
  - Suite of methods emphasizing understanding people rather than technology
- Common mistakes
  - Design by default: “We should do whatever is easiest to implement.”
  - Design by mimicry: “If it’s good enough for Amazon, it’s good enough for us.”
  - Design by fiat: “Because I said so.”

USER XPERIENCE

User experience

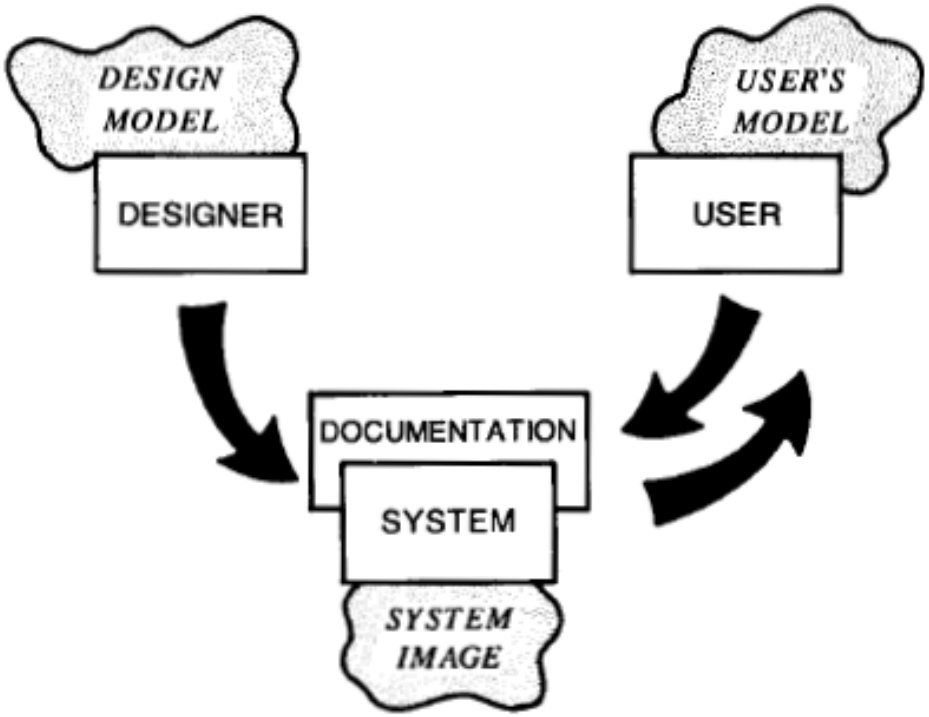
We have a double-edged relationship with the products and services we use. They empower us and frustrate us; they simplify and complicate our lives; they separate us and bring us closer together.

But even though we interact with countless products and services every day, we easily forget that they are made by people, and that someone, somewhere should get the credit when they work well for us—or get the blame when they don't.



J. J. Garrett

User experience



- Design model
  - conceptual model on which the design of the system is based
- System image
  - all aspects of the system that the user experiences
- User's model
  - model that the user develops on the basis of experience with the system

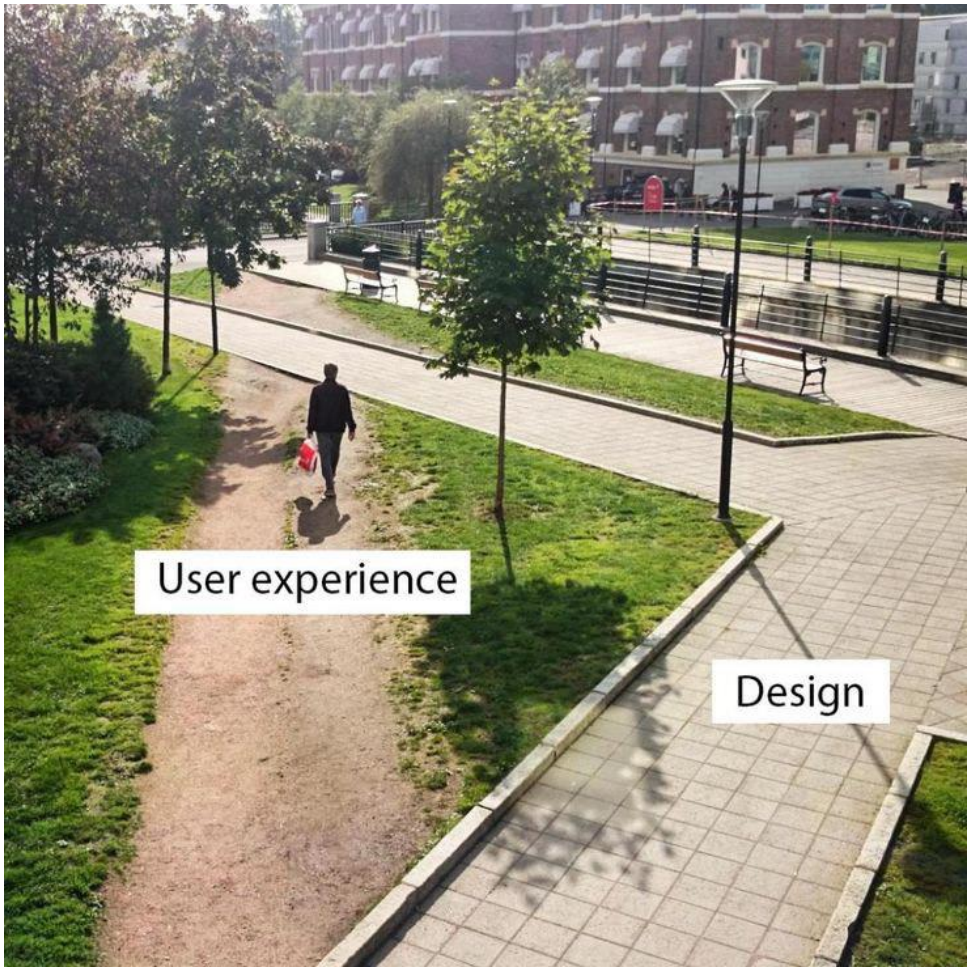
User experience

- User experience is about how a product or a service works on the outside, where a person comes into contact with it
- When most people think about product design they think of:
  - Aesthetic appeal: a well-designed product is one that looks good to the eye and feels good to the touch
  - Functionalities: a well-designed product is one that does what it promises to do (and a badly designed product is one that somehow doesn't, e.g. a printer that constantly jams)

User experience

- User experience design deals also with questions of context
  - Aesthetic design makes sure the button on the coffeemaker is an appealing shape and texture
  - Functional design makes sure it triggers the appropriate action on the device
  - User experience design makes sure the aesthetic and functional aspects of the button work in the context of the rest of the product, asking questions like, “Is the button too small for such an important function?”
  - User experience design also makes sure the button works in the context of what the user is trying to accomplish, asking questions like, “Is the button in the right place relative to the other controls the user would be using at the same time?”
- The more complex a product is, the more difficult it becomes to identify exactly how to deliver a successful experience to the user

Design and user experience

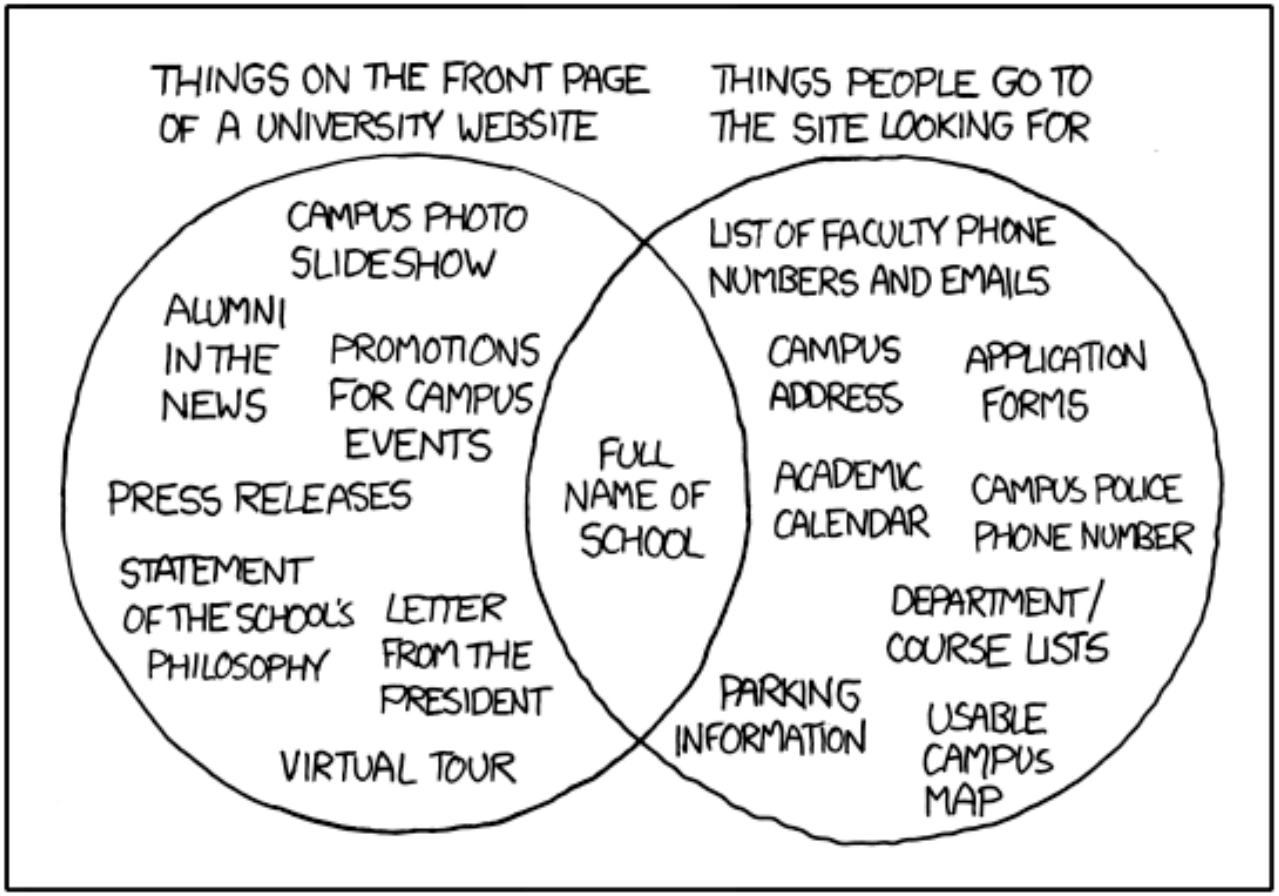


Be Wizard Academy
Usabilità per i siti web

User experience and the Web

- Web sites are complicated, and something funny happens when people have trouble using complicated pieces of technology: they blame themselves
- Regardless of the type of site, in virtually every case, a Web site is a self-service product
  - There is no instruction manual to read beforehand, no training seminar to attend, no customer service representative to help guide the user through the site
  - There is only the user, facing the site alone with only her wits and personal experience to guide her
- Besides, to gain market share against these first-movers, competitors often add more and more content and functionality in hopes of drawing in new customers

User experience and the Web: an example



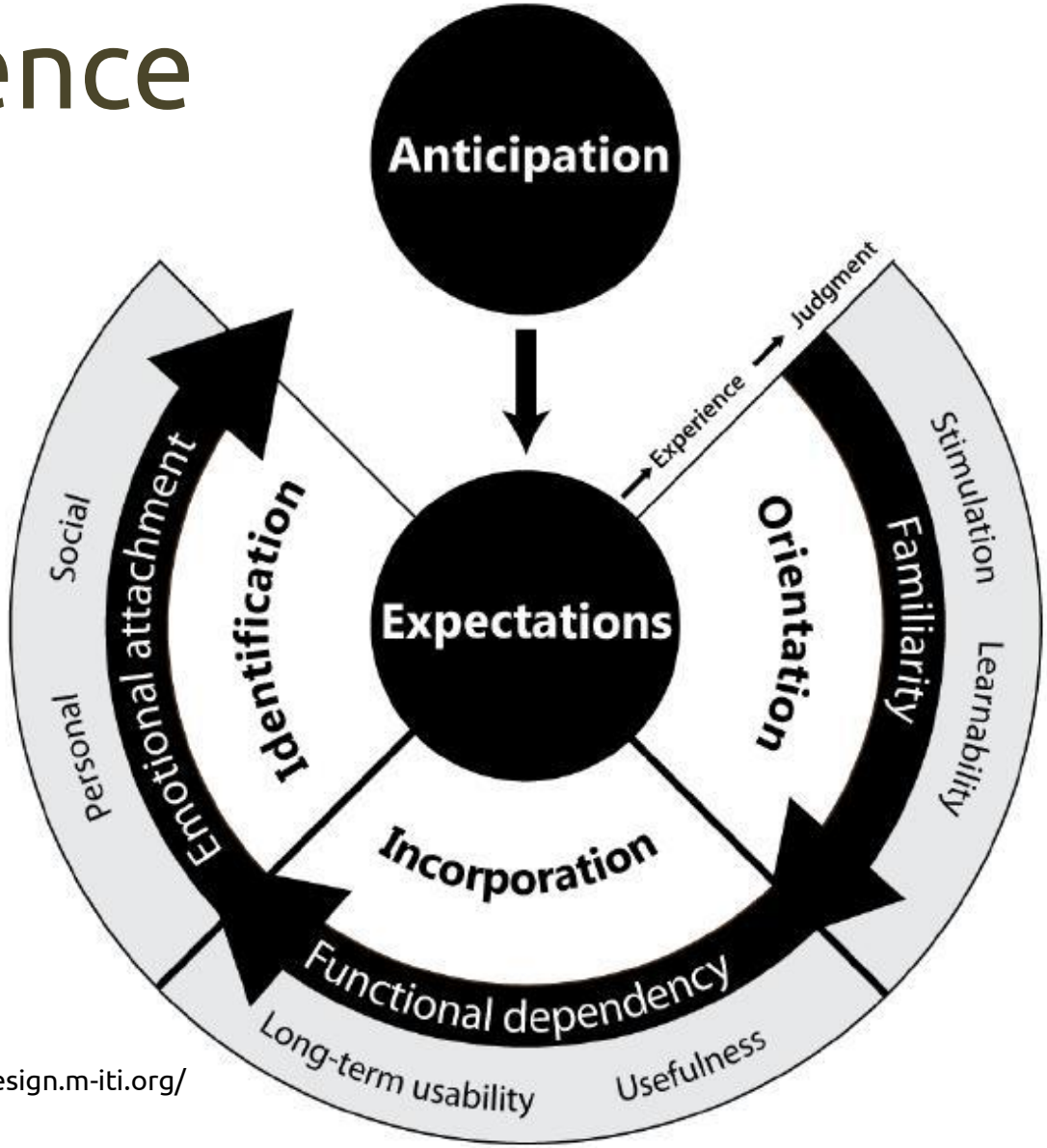
http://xkcd.com/773/

User experience design

- The user experience design process is all about ensuring that no aspect of the user's experience with your product happens without your conscious, explicit intent
  - This means taking into account every possibility of every action the user is likely to take and understanding the user's expectations at every step of the way through that process
- Everything the user experiences should be the result of a conscious decision on your part
  - Realistically, you might have to make a compromise here and there because of the time or expense involved in creating a better solution …
- … But a user-centered design process ensures that those compromises don't happen by accident

User experience

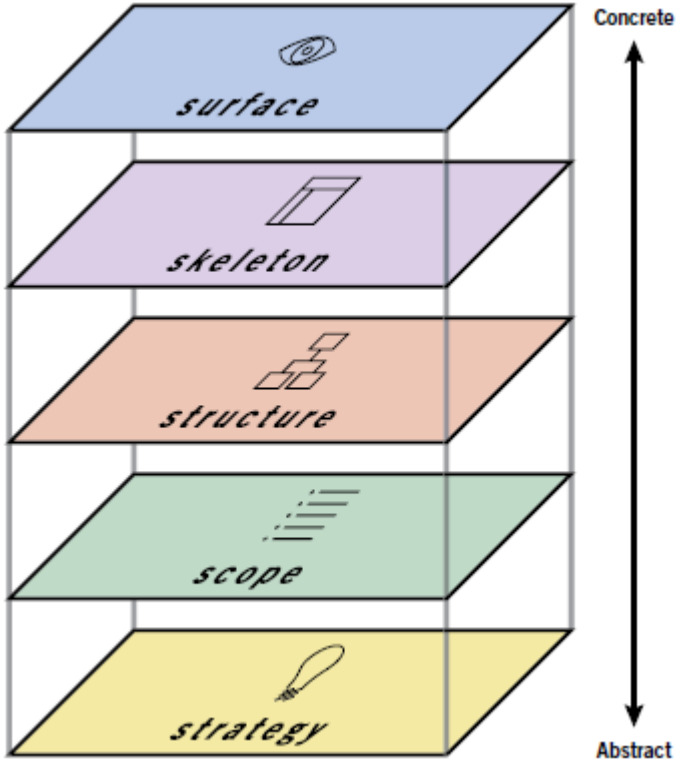
- User experience over time



http://experiencedesign.m-iti.org/

The five plans of web sites user experience

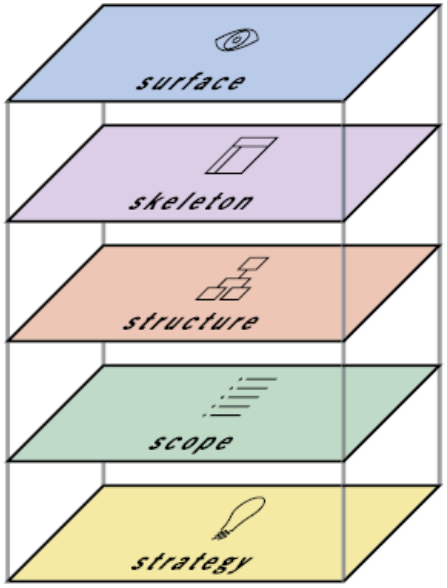
- Conceptual framework user experience problems and tools to solve them
- Each plane is dependent on the planes below



J. J. Garrett

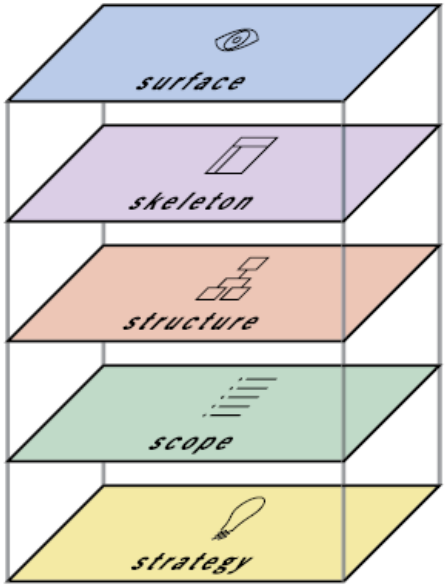
The surface plane

- What will the finished product look like?
  - On the surface you see a series of web pages, made up of images and text
  - Some of these images are things you can click on, performing some sort of function (e.g. taking you to a shopping cart)
  - Some of these images are just illustrations (e.g. a photograph of a product for sale or the logo of the site itself)

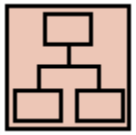


The skeleton plane

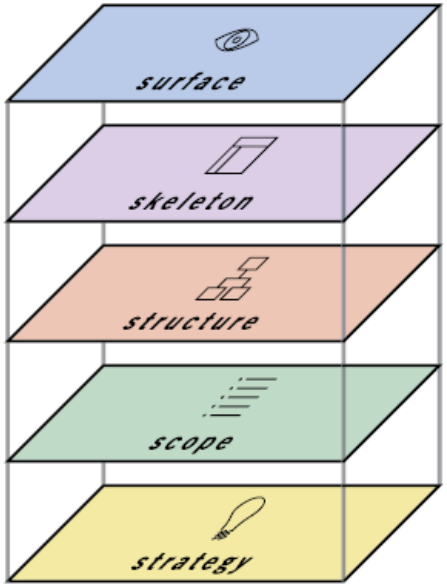
- What components will enable people to use the site?
  - Beneath the surface is the skeleton of the site: the placement of buttons, controls, photos, and blocks of text
  - The skeleton is designed to optimize the arrangement of these elements for maximum effect and efficiency…
  - E.g. … so that you remember the logo and can find that shopping cart button when you need it



The structure plane

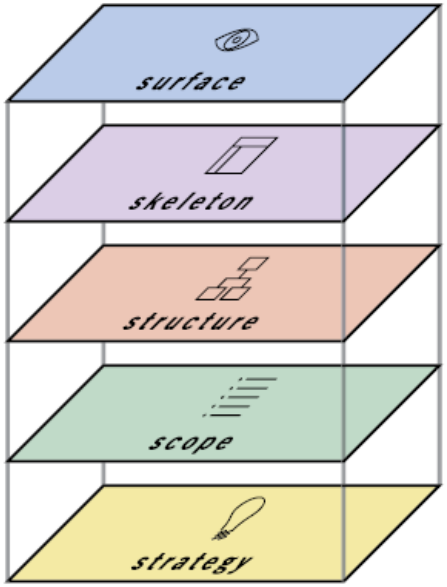


- How will the pieces of the sites fit together and behave?
  - The skeleton is a concrete expression of the more abstract structure of the site
  - The skeleton might define the placement of the interface elements on our checkout page, the structure would define how users got that page and where they could go when they were finished there
  - The skeleton might define the arrangement of navigational elements allowing the users to browse categories of products, the structure would define what those categories are



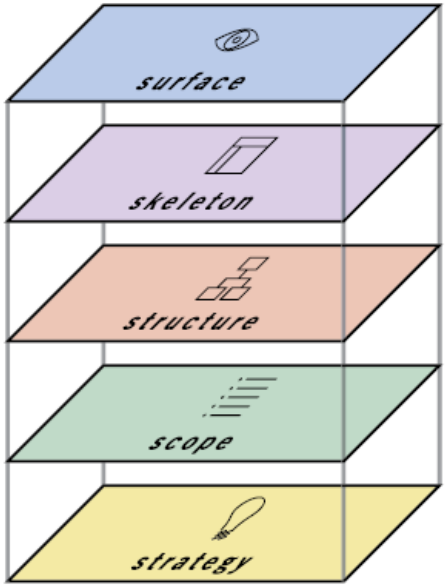
The scope plane

- What feature will the site need to include?
  - The structure defines the way in which the various features and functions of the site fit together
  - What those features and functions are constitutes the scope of the site
  - For example, some commerce sites offer a feature that enables users to save previously used shipping addresses so they can be used again: whether that feature—or any feature—is included on a site is a question of scope



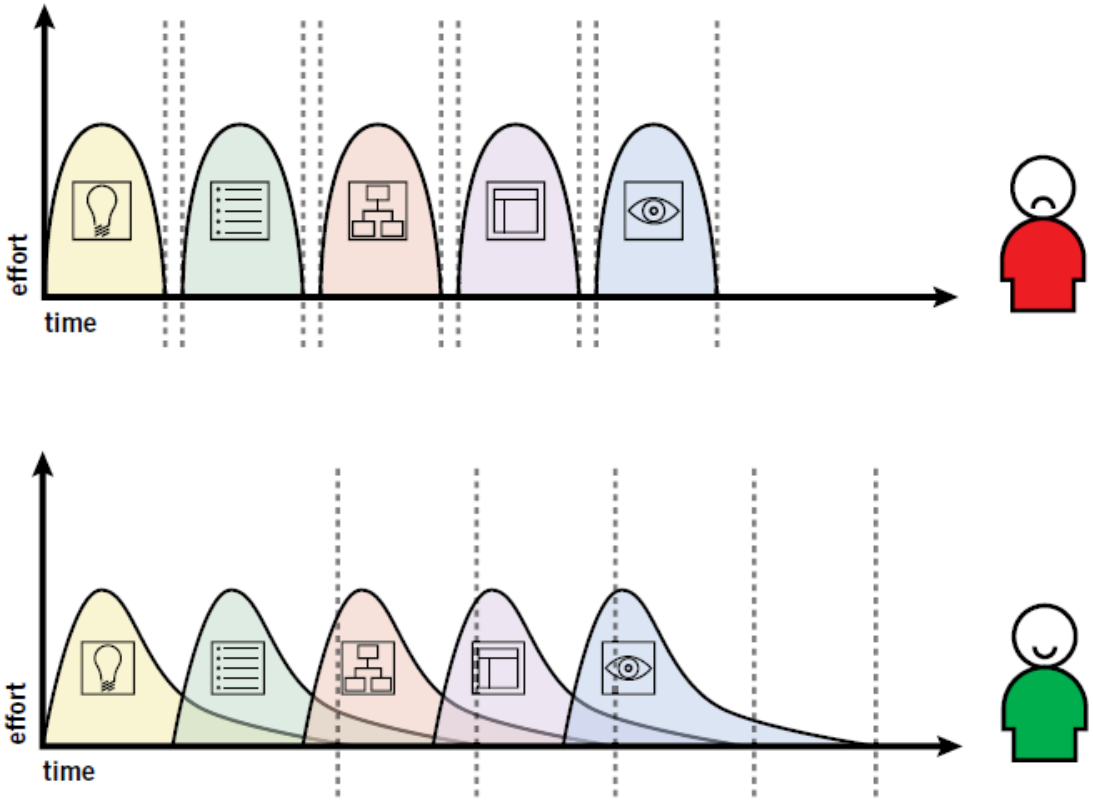
The strategy plane

- What do we want to get out of the site?
- What do our users want?
  - The scope is fundamentally determined by the strategy of the site
  - This strategy incorporates not only what the people running the site want to get out of it but what the users want to get out of the site as well
  - E.g., users want to buy products, and we want to sell them
  - Other objective (such as the role that advertising or content produced by the users plays in the business model, for example) might not be so easy to articulate



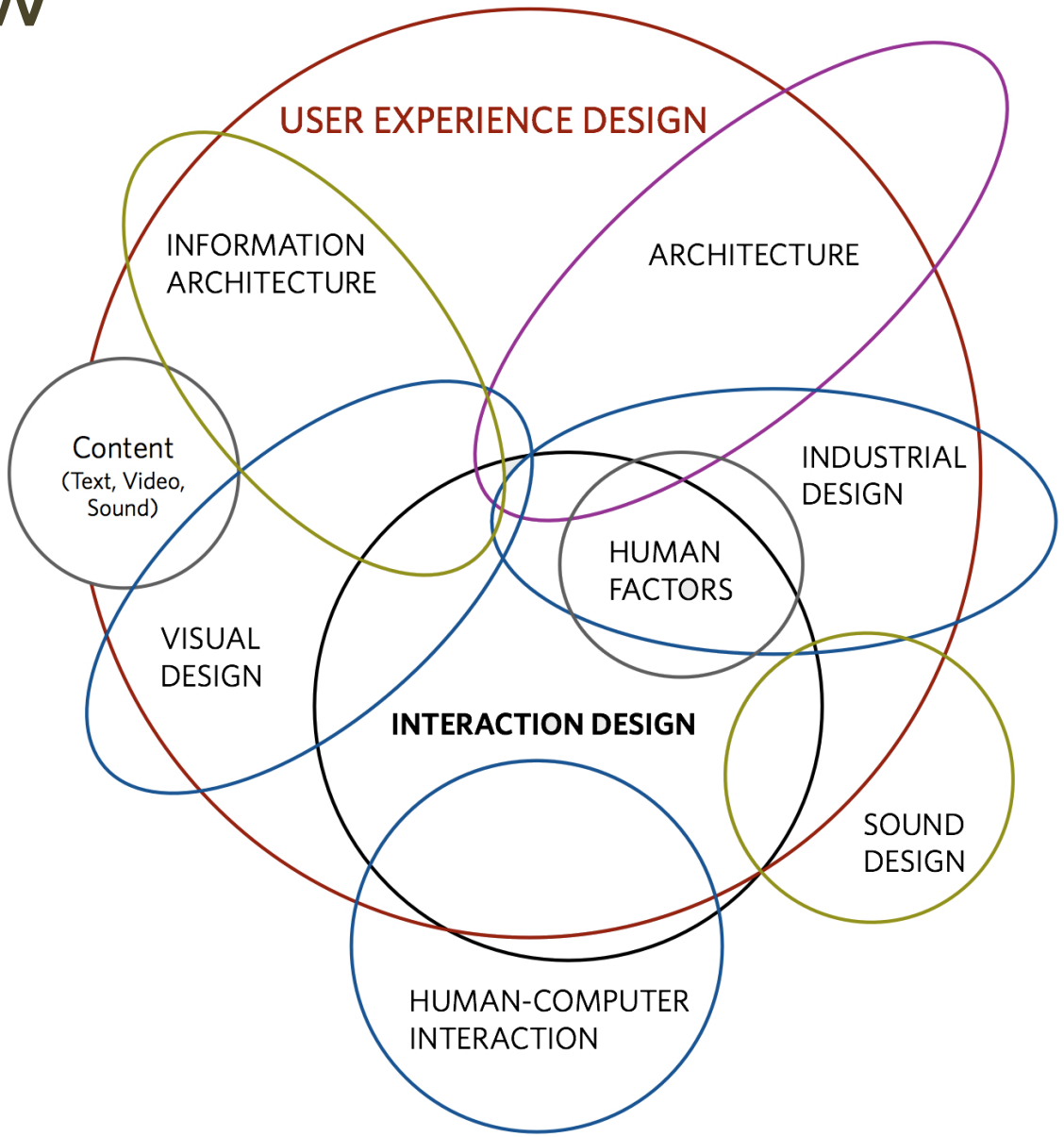
Building from botton to top

- Dependencies among the planes



USER CENTERED DESIGN PROCESS FOR WEB SITES

An overview of design



The web duality

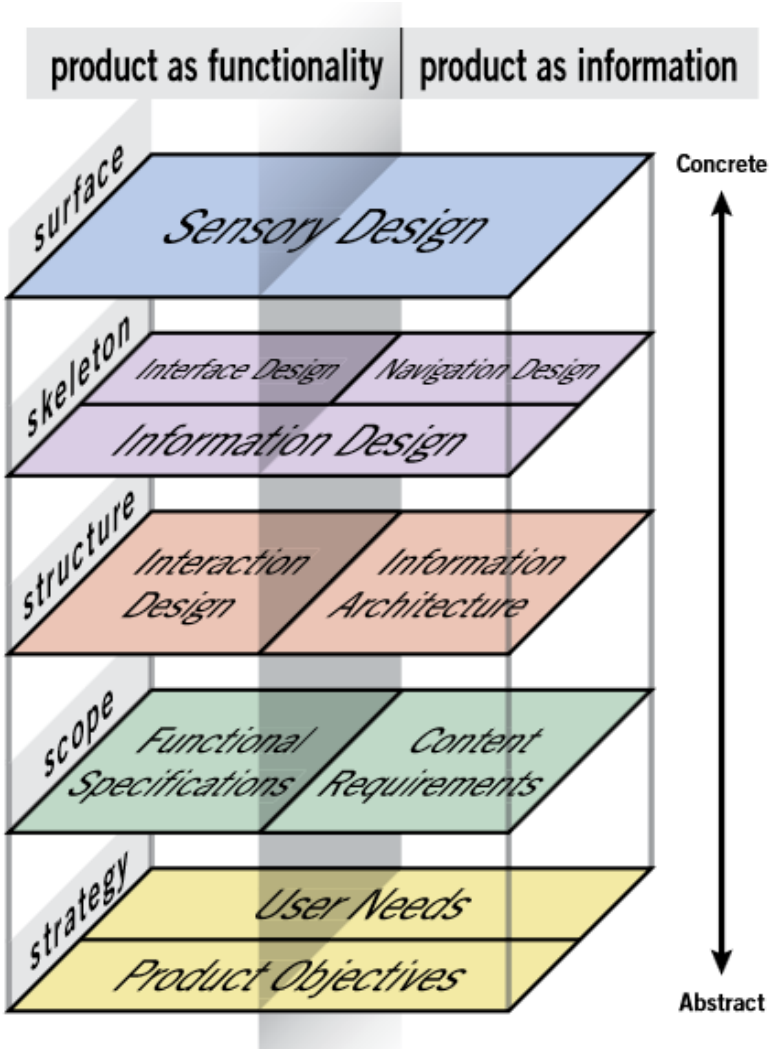
- When the web started, it was all about information
  - People could create documents, and they could link them to other documents
- Then technology advanced and new features were added to Web browsers and Web servers
  - Technology enabled Web sites not only to distribute information but to collect and manipulate it as well
- The Web became more interactive, responding to the input of users in ways that resembled and sometimes moved beyond traditional desktop applications
  - Advent of commercial interests on the Web: wide range of uses, such as electronic commerce, social media, financial services, …

The web duality

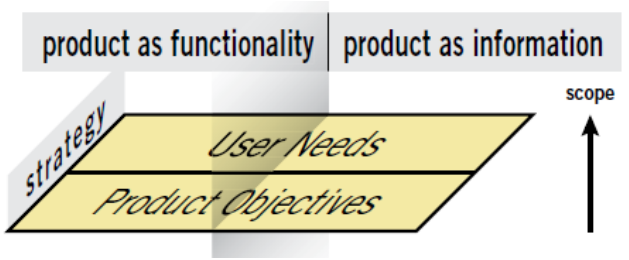
- When the Web user experience community started to form, its members spoke two different languages
  - One group saw every problem as an application design problem, and applied problem-solving approaches from the traditional desktop and mainframe software worlds
  - The other group saw the Web in terms of information distribution and retrieval, and applied problem-solving approaches from the traditional worlds of publishing, media, and information science
- Questions:
  - «where» is a Social Network Site?
  - «where» is YOUR Social Network Site?

The elements of user experience

- A framework for design
- Implications for the design process

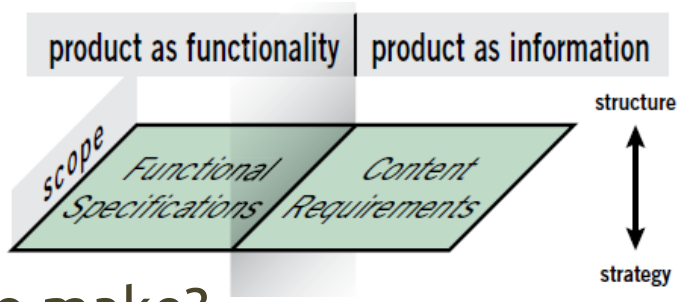


The strategy plane



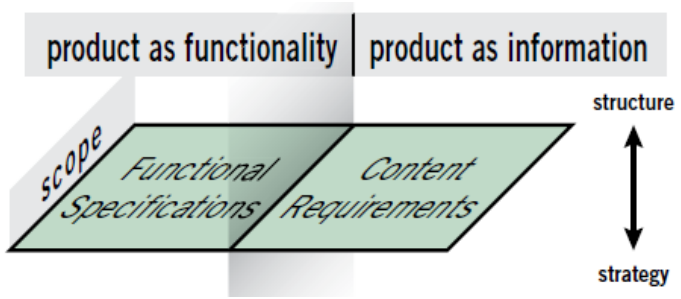
- The most common reason for the failure of a Web site is not technology, and it is not user experience
- Web sites most often fail because nobody bothered to answer two very basic questions:
  - What do we want to get out of this product?
  - What do our users want to get out of it?
- Answer should be explicit, grounded and well documented
- Product (site) objectives
  - Come from inside the organization (e.g., business, creative, …)
  - Business goals, fidelization, creative goals, success metrics, …
- User needs
  - Objectives imposed on the product from outside
  - Identified through user research
  - Personas

The scope plane



- Main question: what are we going to make?
- Importance of clearly identified and concrete requirements
  - Lots of features sound like good ideas, but they don't necessarily align with the strategic objectives
- Functional specifications (or requirements)
  - What will be the «feature set» of the software product, with detailed description
- Content requirements
  - What information needs to be included in the site under development
- Content requirements often have functional implications
  - Often, the functionalities needed in the site depend on the nature of the content to be managed
- Functional requirements have content implications
  - Will there be instructions on the preferences configuration screen? How about error messages? Someone has to write them…

The scope plane



- Some requirements apply to the product as a whole
  - E.g., some technical requirements such as supported browsers and operating systems
- Other requirements apply only to a specific feature
  - Short description of a single feature
- The level of detail depends on the specific scope of the project
- The most productive source for functional and content requirements are users and stakeholders themselves, but need to be processed and reiterated
- Requirements that come out of the process fall into three general categories
  - Things people say they want
  - Things people say they want that are not the things they actually want
  - Things people don't know they want

Functional specifications

- Can change during implementation
- Very important: clarity and accuracy
- Positive
  - “The system will not allow the user to purchase a kite without kite string.” vs “The system will direct the user to the kite string page if the user tries to buy a kite without string.”
- Specific
  - “The most popular videos will be highlighted.” vs “Videos with the most views in the last week will appear at the top of the list.”
- No subjective language
  - “The site will have a hip, flashy style.” vs “The look of the site will conform to the company branding guidelines document.”

Content requirements

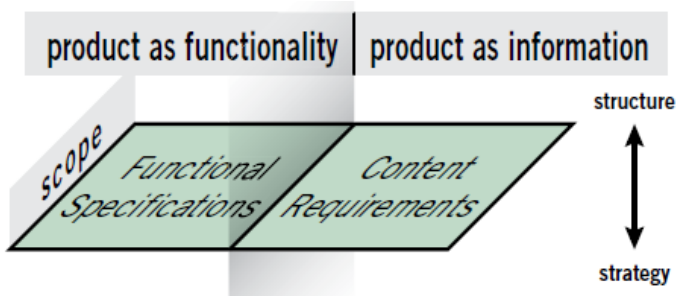
- Identifying all the content types associated with a feature (text, images, videos, …)
- Focus on content and not on format
  - E.g., FAQ comes from user interaction, not from the content provider
- The expected size of each of your content has a huge influence on the future user experience decisions
  - Word count for text features, pixel dimensions for images or video (e.g., use thumbnail images?), file sizes for downloadable documents, ...
- Identify the responsible for each content element as soon as possible
- Content requires constant maintenance
  - Approaching content as if you can post it and forget it leads to a site that, over time, does an increasingly poor job of meeting user needs
- For every content element, identify how frequently it will be updated
- If your site has to serve multiple audiences with divergent needs, knowing which audience a piece of content is intended for can help you make better decisions about how to present that content

Requirements (what is wanted)

- The first stage is establishing what exactly is needed
- What is currently happening? What is the goal?
- Techniques used in HCI: interviewing people, videotaping them, looking at the documents and objects that they work with, observing them directly, ethnography (a form of observation derived from anthropology)…

The process of interaction design

- Analysis
  - The results of observation and interview need to be ordered in some way to bring out key issues and communicate with later stages of design
  - Techniques: scenarios (rich stories of interaction), task models, task analysis methods
  - These techniques can be used both to represent the situation as it is and the desired situation
- Results:
  - Functional specifications
  - Content requirements



Formalizing requirements

- The initial vision and user inputs must be “distilled” into a set of requirements
- Strategic choices: what is in, what is out
- Describes what the system does, and the external constraints
- Might be used as a “specification contract”
- It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification
- Requirements engineering
  - The process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed

Levels of requirement

- User requirements
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints
  - Written for customers
- System requirements (or developer requirements)
  - A structured document setting out detailed descriptions of the system's functions, services and operational constraints
  - Defines what should be implemented so it may be part of a contract between client and contractor

Example

User requirement definition

The software must provide a means of representing and accessing external files edited by other tools

System requirements specification

1.1 The user should be provided with facilities to define the type of external files

1.2 Each external file type may have an associated tool which may be applied to the file

1.3 Each external file type may be represented as a specific icon on the user's display

1.4 Facilities should be provided for the icon representing an external file type to be defined by the user

1.5 When a user selects an icon representing an external file the effect of that selection is to apply the tool associated with the external file type to the file represented by the selected icon

Types of requirements

- Functional requirements (FR)
  - Statements about services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations
- Non-functional requirements (NFR)
  - Aka quality requirements
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.

Functional Requirements (FR)

- What the system does
- What functions it offers to its users
- A long list of “local” features (easy to identify a small portion of the system that delivers that function)

- Don’t care how they will be implemented (yet)

Examples

- FR3.1: The user must be able to activate and de-activate the wake-up service. This decision will be applied until the user changes it again.
- FR3.2: The user must be able to silence the wake-up service just for the next day. Service will resume automatically on the following day.
- FR4.4: The user must be able to set up an “ad hoc” wake-up call, that will run only once, will not be remembered, and will have specific settings
- FR4.4.1: The user may configure the settings of any already defined “ad hoc” call
- FR4.4.2: The user may configure the default settings for (to be created) “ad hoc” calls

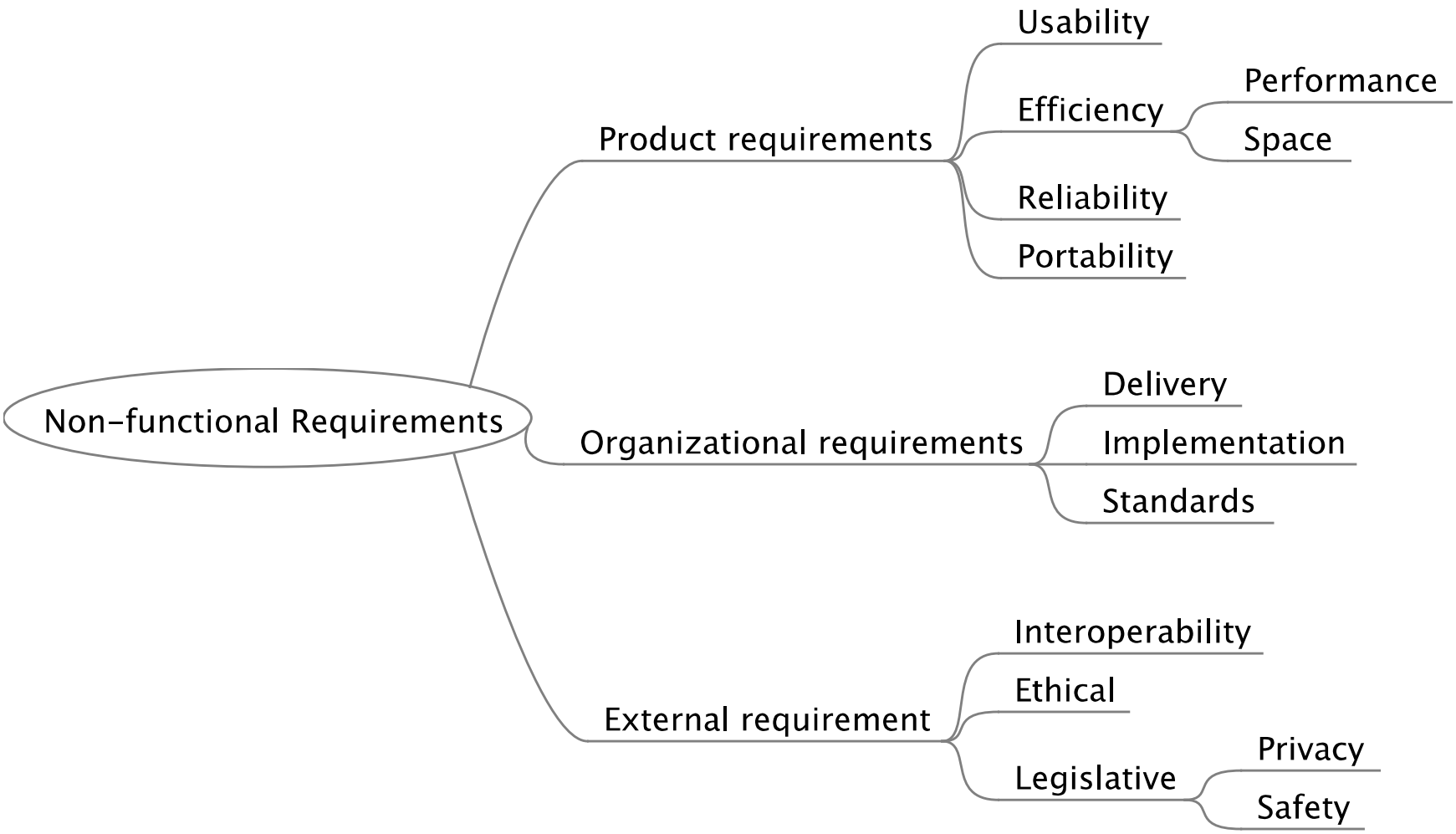
Non-functional requirements (NFR)

- Define system properties and constraints e.g. reliability, response time and storage requirements.
  - Constraints are I/O device capability, system representations, Supported devices, Usability, Language, etc.
- Process requirements may also be specified mandating a particular set of tools, programming language or development method
- Non-functional requirements may be more critical than functional requirements: if these are not met, the system is useless.

Pervasiveness in NFR

- NFR are usually “general” and cannot be localized to a single spot in system implementation
- Every function, in every module, in every screen, in every device, …. must guarantee that no NFR is broken

Non-functional requirements



Examples

- NFR1: The mobile interfaces must be compatible with iOS (8.0 and later), Android (4.2 and later)
- NFR2: The system will be localized in many languages (default: English)
- NFR18: The system should work, in reduced conditions, even if the user mobile device is switched off or disconnected
- NFR3: The web interfaces will be compatible with browsers … version ….
- NFR4: The web interfaces will be “responsive”, and will adapt to screen resolutions from 800x600 to 1920x1080

Functional vs non-functional

Functional Requirement	Non-functional Requirement
Defines all the services or functions required by the customer that must be provided by the system	Defines system properties (e.g. reliability, response time, storage requirements, …) and constraints (e.g. I/O device capability, system representations, …)
Describes what the software should do	Describes how the software will do (what it should do)
Related to business	Related to improving the performance of the business
Easy to test	Difficult to test
Related to individual system features	Related to the system as a whole
Failure to meet a functional requirement may degrade the system	Failure to meet a non-functional requirement may make the whole system unusable

Best requirements

- Complete (express a whole idea or statement)
- Correct (technically and legally possible)
- Clear (unambiguous and not confusing)
- Verifiable (it can be determined that the system meets the requirement)
- Necessary (should support one of the project goals)
- Feasible (can be accomplished within cost and schedule)
- Prioritized (tracked according to business need levels)
- Consistent (not in conflict with other requirements)
- Traceable (uniquely identified and tracked)
- Modular (can be changed without excessive impact)
- Design-independent (do not pose specific solutions on design)

Complete

- Express a whole idea or statement
  - Should describe completely the user task and the information required to support the task
  - Define response to all possible inputs (both correct and incorrect)
  - Define all terms and unit of measure
  - Focusing on system functionality instead of user needs to be accomplished may lead to incomplete requirements
- Example

‘We must be able to change an employee’s profile information’

‘We must be able to change the employees last name, first name, middle initial, street address, city, state, zip code, marital status’

Correct

- Technically and legally possible
  - The requirements should be appropriate to meet the goals of the project and accurately describe the user's expectations of the functionality
  - Customer or users can determine if the requirement correctly reflects their actual needs
- Example

'Employees only change their entry when their address or their marital status changes'

'Employees may change their entry in the payroll system by providing the appropriate legal proof of the change. The change may come with a change in marital status or address'

Clear

- Unambiguous and not confusing
  - Requirements should be written so that all readers will arrive at a single, consistent interpretation
  - Clear to those who create it and to those who use it
  - Ambiguous requirements can result in the wrong system being developed and may not be found during testing due to the incorrect interpretation of the requirements
- Example

‘Employees are not allowed to work for more than 80 hours in one week’

‘Employee time worked: the time worked is recorded in hours, the smallest increment recorded is .25 of an hour. If an employee reports more than 80 hours in a 7 day period, a warning is provided to the supervisor and the payment is held for approval.’

Verifiable

- It can be determined whether the system meets the requirement
  - Each requirement should be testable and verifiable
  - There exists some finite cost-effective process with which a person or machine can check that the software product meets the requirement
  - Ambiguous requirements are not verifiable
- Example

‘The system should be easy to use’

‘A novice user must be able to add a new employee to the payroll system within 10 minutes’

Necessary

- Should support one of the project goals
  - Related to specific and meaningful goals
- Example

‘We should be able to enter the employee eye colour’

Why is this requirement necessary?

Feasible

- It can be accomplished within cost and schedule
  - The business analyst must be sure that all requirements are technologically possible for a reasonable cost
- Example

‘The system should automatically be updated when the government changes the law’

Although this requirement may be technologically feasible, it would involve a complex interface (and likely new government system) with an outside organisation which would be very expensive and difficult to negotiate. Is it a critical requirement?

Prioritized

- Tracked according to business need levels
  - Each requirement should be prioritised
- Most organisations use the MoSCoW method for prioritisation
  - Must Have: the system must meet this requirement for the end product to be considered a success
  - Should Have: the system should have this requirement for it to solve the main business problem
  - Could Have: it would be good to include this requirement to ensure maximum benefit
  - Would Have: this is a nice to have requirement which the business could do without if necessary

Consistent

- Not in conflict with other requirements
  - No subset of requirements is in conflict
  - Logical and temporal consistence
- Traceable: uniquely identified and tracked
  - Backward: explicitly referencing source in earlier documents
  - Forward: unique name or reference number

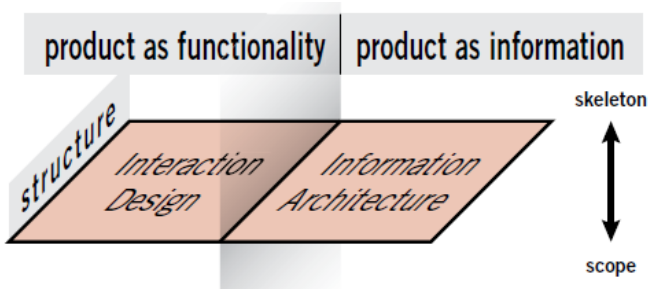
Modular and design-independent

- Modular: can be changed without excessive impact
  - Structure and style such that any changes can be made easily, completely, and consistently while retaining the structure and style
  - Well structured, non redundant, separate requirements
- Design-independent: do not pose specific solutions on design

Prioritizing requirements

- Collecting ideas for possible requirements is not (too…) hard
- The tricky part is sorting out what features should be included in the scope of the project
- Evaluation based on
  - Whether requirements fulfill the strategic goals (both product objectives and user needs)
  - How feasible will it be to actually implement them (technical feasibility, organizational constraints, availability of resources, time constraints)
- Any feature suggestion not in line with the project strategy is, by definition, out of scope …
- … but if a suggested feature that falls outside the scope doesn't fit any of the types of constraints and still sounds like a good idea, you may want to reexamine some of the strategic objectives

The structure plane

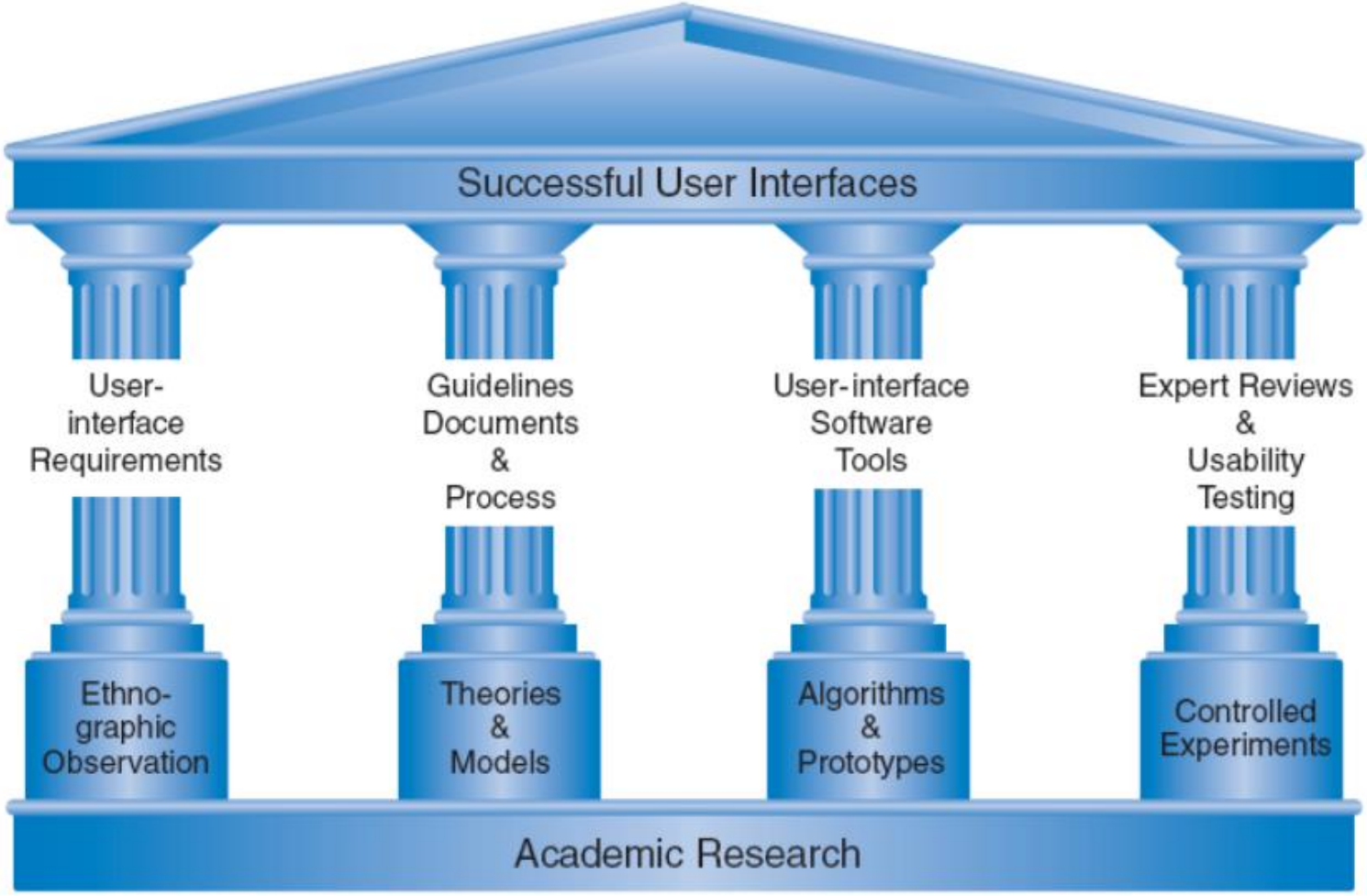


- Objective: starting from the prioritized requirements, to develop the conceptual structure of the site
- Interaction design
  - Development of application flows to facilitate user tasks
- Information architecture
  - Structural design of the information space to facilitate intuitive access to content
- Emphasis on defining patterns and sequences in which options will be presented to users
  - Interaction design: options involved in performing and completing tasks
  - Information architecture: options involved in delivering information to a user

Interaction design

- Objective: to describe possible user behavior and to define how the system will accommodate and respond to that behavior
- Traditionally, software programmers try to build a system in the way that was most technically efficient but …
- …the approach that works best for the technology is almost never the approach that works best for the person using it
  - Perception that “software is complicated, confusing, and hard to use”
- Conceptual models, often based on real-world metaphors
  - E.g., shopping cart component in an e-commerce site

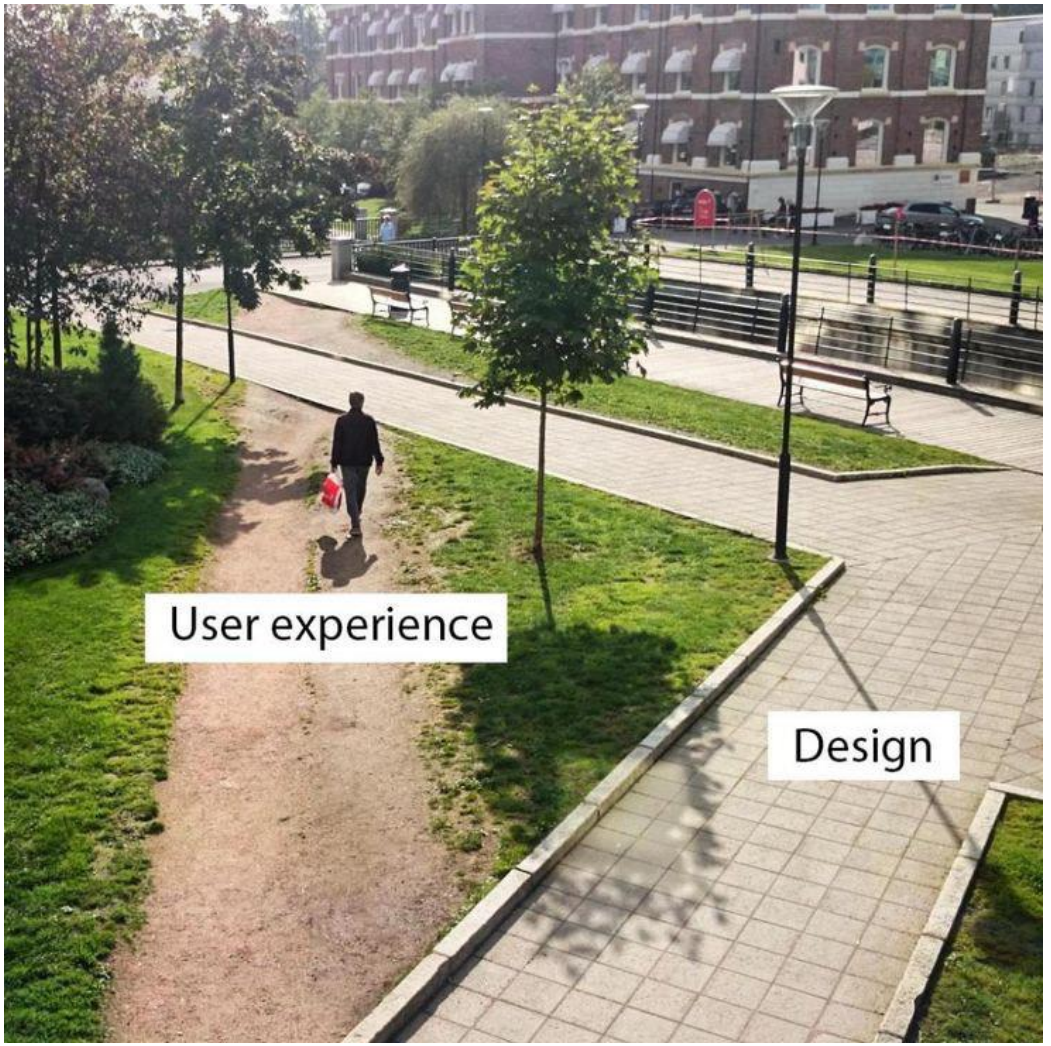
The pillars of interaction design



Design for usability

- One of the central problems that must be solved in a user-centered design process is how to provide designers with the ability to determine the usability consequences of their design decisions
- Design rules, classified along two dimensions: authority and generality
  - Principles: abstract design rules, with high generality and low authority
  - Standards: specific design rules, high in authority and limited in application
  - Guidelines: lower in authority and more general in application
- Can be supported by psychological, cognitive, ergonomic, sociological, economic or computational theory, which may or may not have roots in empirical evidence

Design and user experience



Be Wizard Academy
Usabilità per i siti web

Principles to support usability

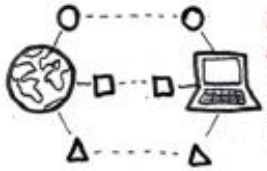
- Three main categories
  - Learnability: the ease with which new users can begin effective interaction and achieve maximal performance
  - Flexibility: the multiplicity of ways in which the user and system exchange information
  - Robustness: the level of support provided to the user in determining successful achievement and assessment of goals

Golden rules and heuristics

- Nielsen's ten heuristics (mainly for evaluation)
- Shneiderman's eight golden rules
- Norman's seven principles

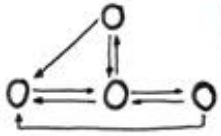
Visibility of system status

Give the users appropriate feedback about what is going on.



Match between system and the real world

Use real-world words, concepts and conventions familiar to the users in a natural and logical order.



User control and freedom

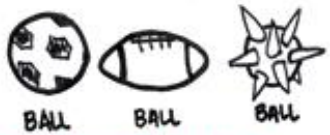
Support undo, redo and exit points to help users leave an unwanted state caused by mistakes.

Error prevention

Prevent problems from occurring: eliminate error-prone conditions or check for them before users commit to the action.

Aesthetic and minimalist design

Don't show irrelevant or rarely needed information since every extra elements diminishes the relavance of the others.



Consistency and standards

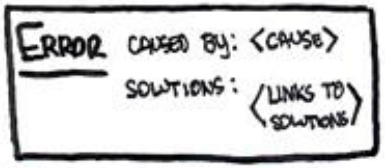
Follow platform conventions through consistent words, situations and actions.

Recognition rather than recall

Make objects, actions, and options visible at the appropriate time to minimize users' memory load and facilitate decisions.

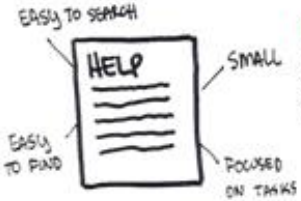
Flexibility and efficiency of use

Make the system efficient for different experience levels through shortcuts, advanced tools and frequent actions.



Help users recognize, diagnose, and recover from errors

Express error messages in plain language (no codes) to indicate the problem and suggest solutions.

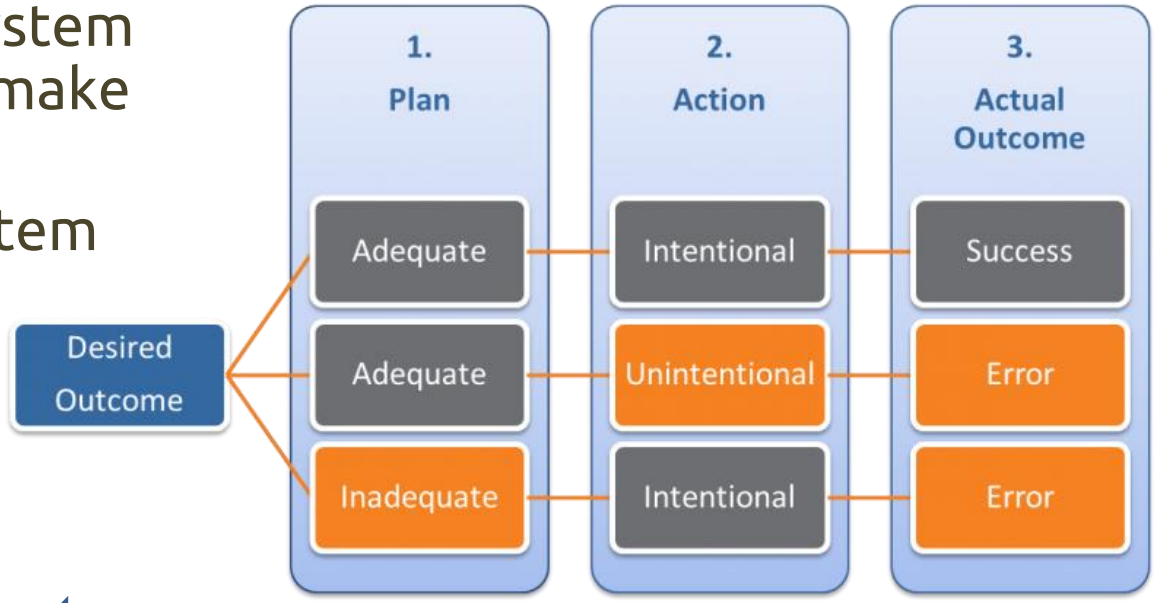


Help and documentation

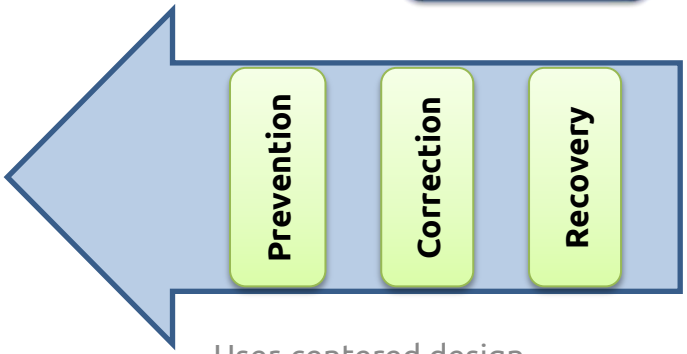
Make necessary help and documentation easy to find and search, focused

Interaction design

- Error handling
  - What does the system do when people make mistakes?
  - What can the system do to prevent those mistakes from happening in the first place?



- Three layers

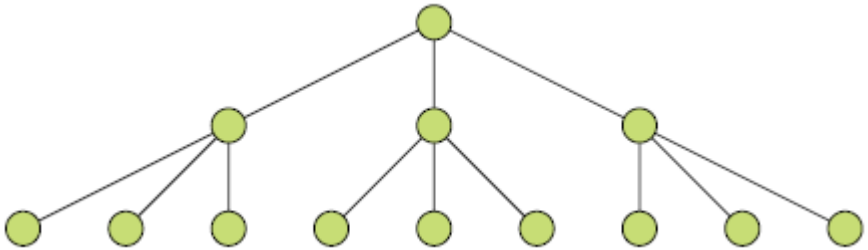


Information architecture

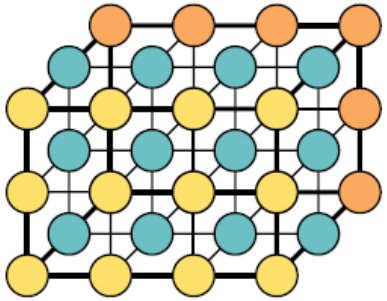
- Should consider how people cognitively process information to make sense of it
- Creating organizational and navigational schemes that allow users to move through site content efficiently and effectively
- Related to information retrieval: the design of systems that enable users to find information easily
- Most commonly, information architecture problems require creating categorization schemes
  - Top-down approach: involves creating the architecture directly from an understanding of product objectives and user needs
  - Bottom-up approach: derives categories and subcategories based on an analysis of the content and the functional requirements

Architectural approaches

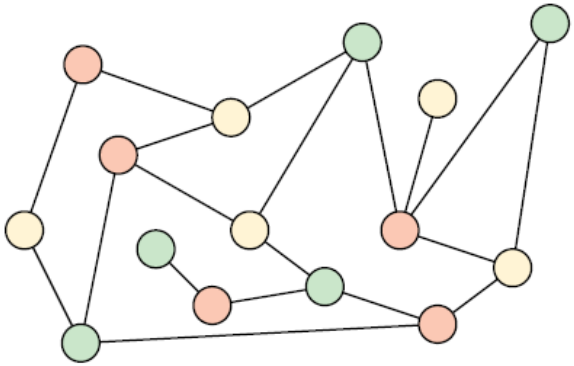
- Hierarchical structure



- Matrix structure



- Organic structure



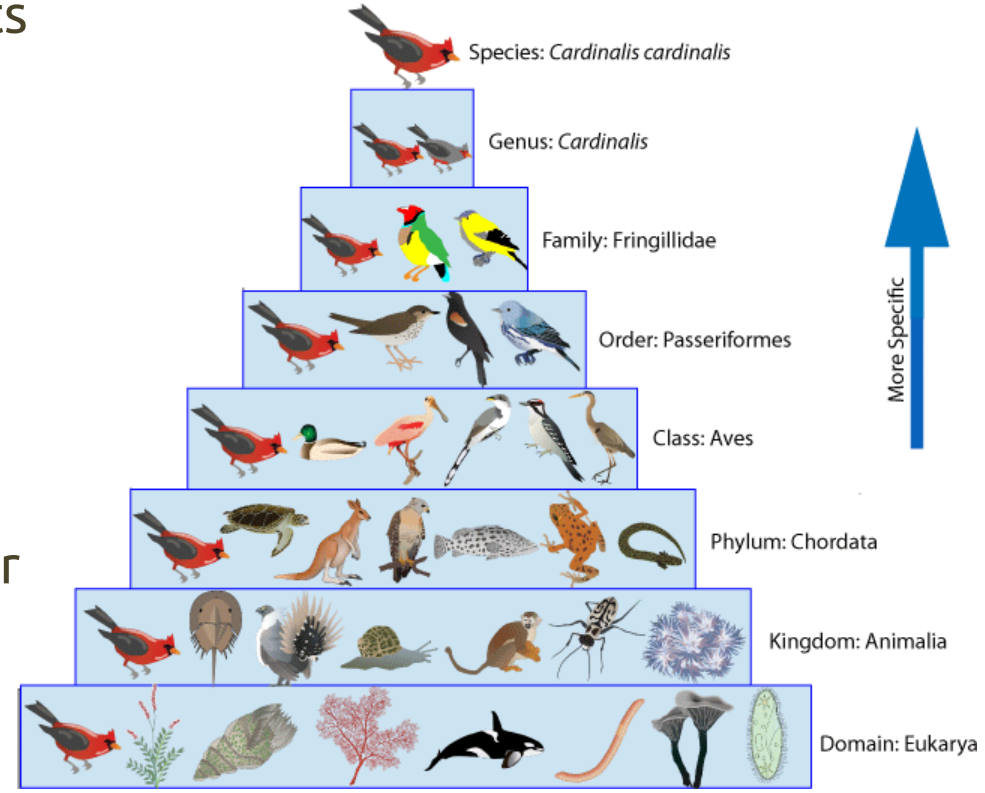
- Sequential structure

Language and metadata

- Even if the structure is a perfectly accurate representation of the way people think about your subject matter, your users won't be able to find their way around the architecture if they can't understand your nomenclature: the descriptions, labels, and other terminology the site uses
  - Subject-based classification: controlled vocabularies, taxonomies, thesauri, ontologies
  - Metadata
- Talking to users and understanding how they communicate is the most effective way to develop a system of nomenclature that will feel natural to them

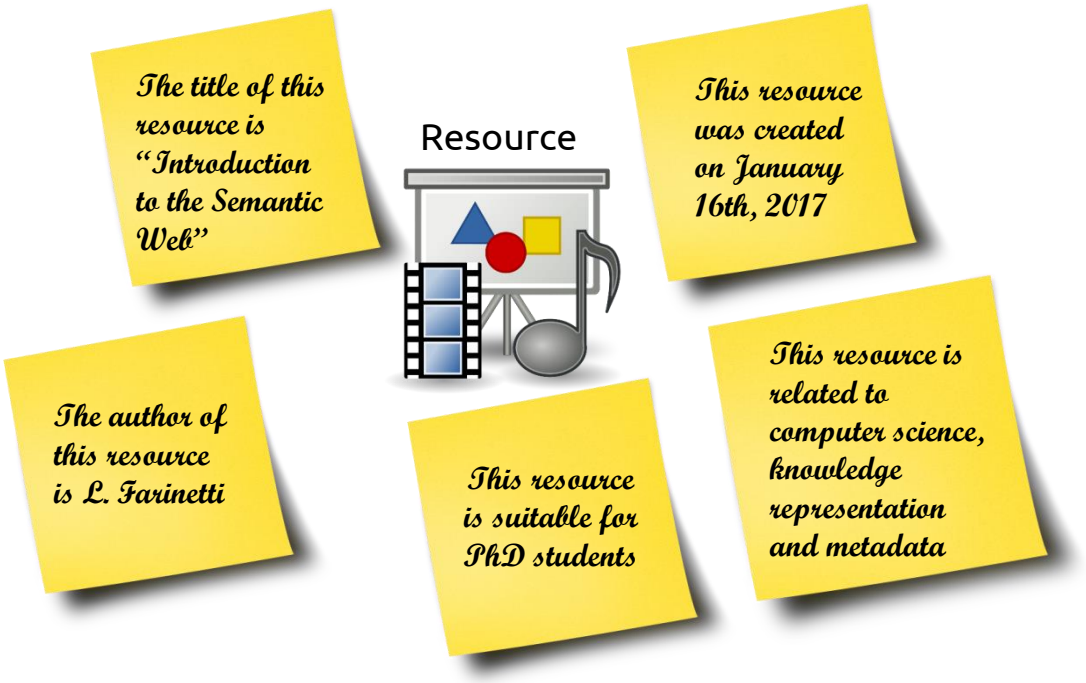
Subject-based classification

- Controlled vocabulary: a closed list of named subjects
- Taxonomy: arranges the terms in the controlled vocabulary into a hierarchy
- Thesaurus: extends taxonomies with more relationship type (BT, USE, TT, RT, …)
- Ontology: extends the other classification approaches because has open vocabularies and open relationship types



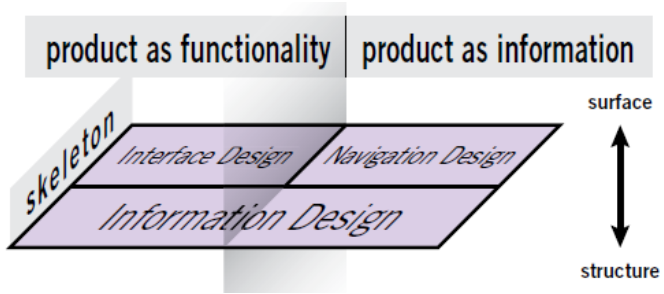
Metadata

- Resource and description



- Good metadata can provide a faster and more reliable way for users to find information on your site than a basic full-text search engine

The skeleton plane



- Structure plane: the large-scale issues of architecture and interaction
- Skeleton plane: smaller scale of individual components and their relationships
- Interface design
  - Design of the interface elements (buttons, input fields, …) to facilitate user interaction with functionalities
- Navigation design
  - Design of the interface elements to facilitate the user navigation through the information architecture
- Information design
  - Presentation of information for effective communication

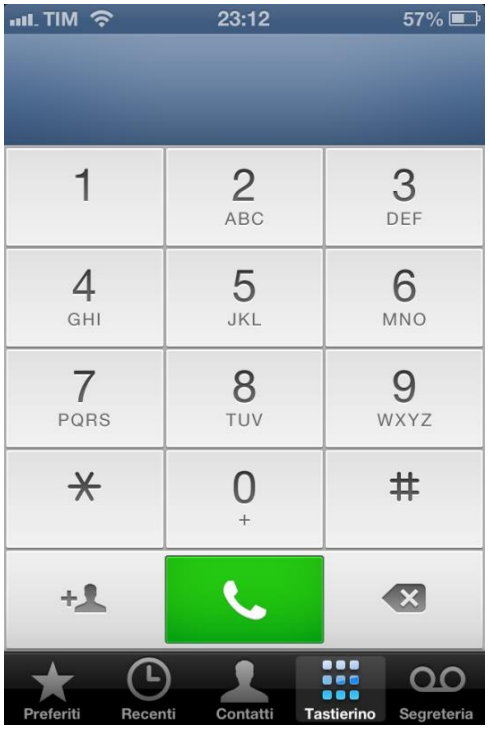
The skeleton plane

- The three elements are closely bound together but…
  - … if it involves providing users with the ability to do things, it's interface design
  - … if it involves providing users with the ability to go places, it's navigation design
  - …if it involves communicating ideas to the user, it's information design
- Habit and reflex are the foundation for much of our interaction with the world
  - Importance of consistency (with other products and within the product itself)
- Page layout is where information design, interface design, and navigation design come together to form a unified, cohesive skeleton
  - Wireframes (page schematics) or prototypes!!!

Interface design

- All about selecting the right interface elements for the task the user is trying to accomplish and arranging them on the screen in a way that will be readily understood and easily used
- Tasks will often stretch across several screens, each containing a different set of interface elements for the user to contend with
  - Which functions end up on which screens is a matter of interaction design down in the structure plane
  - How those functions are realized on the screen is the realm of interface design
- Successful interfaces are those in which users immediately notice the important stuff
  - Unimportant stuff is not there at all (possibly)
- A well-designed interface recognizes the action users are most likely to take and makes those interface elements easiest to access and use

Interface conventions



Phone

Calculator

- How is the ATM numeric keyboard?

Interface conventions

- Slowly, but they change

Checkboxes are independent
So they can come in groups

Or stand alone

Radio buttons
Come in groups
And are used to make
Mutually exclusive selections

Hamburgher menu



Navigation design

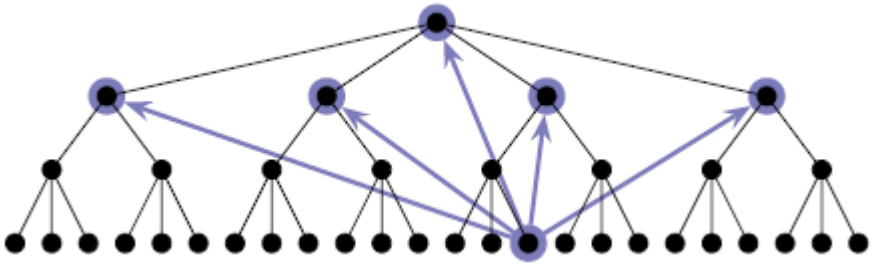
- Designing navigation for the Web looks simple: put links on every page that allow users to get around on the site…
- … But in reality it is really complex: three simultaneous goals
  - It must provide users with a means for getting from one point to another on the site: selection of links to facilitate users' behavior
  - It must communicate the relationship between the elements it contains: are some links more important then others?
  - It must communicate the relationship between its contents and the page the user is currently viewing: what this links has to do with the present content?
- It is of vital importance that every page of a Web site communicate clearly to users where they are on the site and where they can go

Navigation design

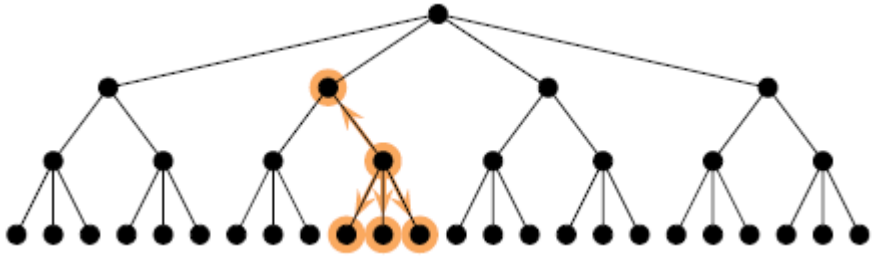
- It’s hard to know how (or how much) people keep the structure of Web sites in their heads: without that knowledge, the best approach is to assume that users carry no knowledge with them from page to page
  - Besides, search engines index the sites, so any page can be an entry point
- Multiple navigation systems should be provided in the site
- Navigation elements that appear throughout a site are called persistent

Navigation systems

- Global navigation
  - Provides access to the set of key points that users might need to get from one end of the site to the other



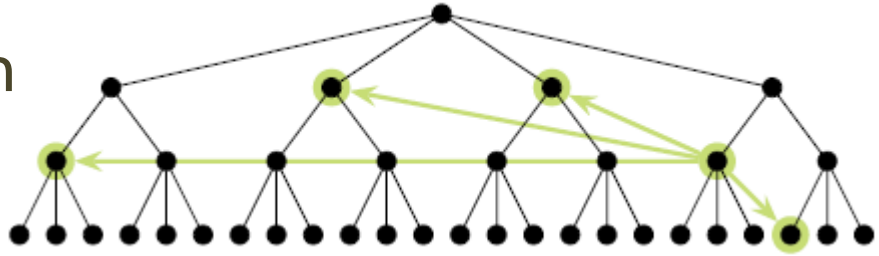
  - Example: navigation bars linking to all the main sections of a site
- Local navigation
  - Provides access to what's "nearby" in the architecture



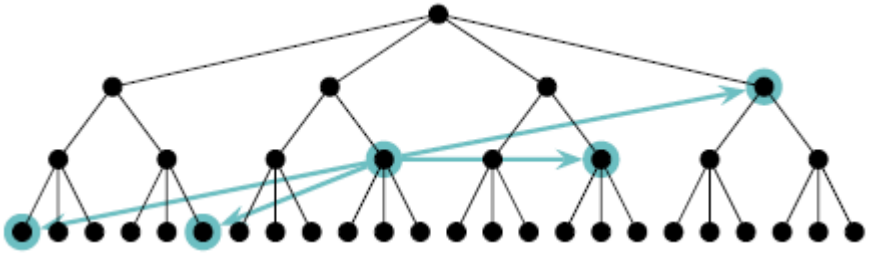
  - Example: in a strictly hierarchical architecture, local navigation might provide access to a page's parent, siblings, and children

Navigation systems

- Supplementary navigation
  - Provides shortcuts to related content that might not be readily accessible through the global or local navigation
  - Example: in faceted classification allows users to shift the focus of their exploration of the content without starting over at the beginning

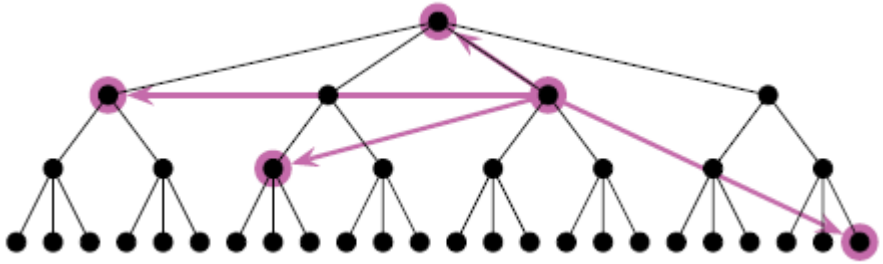


- Contextual navigation (or inline navigation)
  - Is embedded in the content of the page
  - Example: a hyperlink within the text of a page
  - Often underutilized or misutilized



Navigation systems

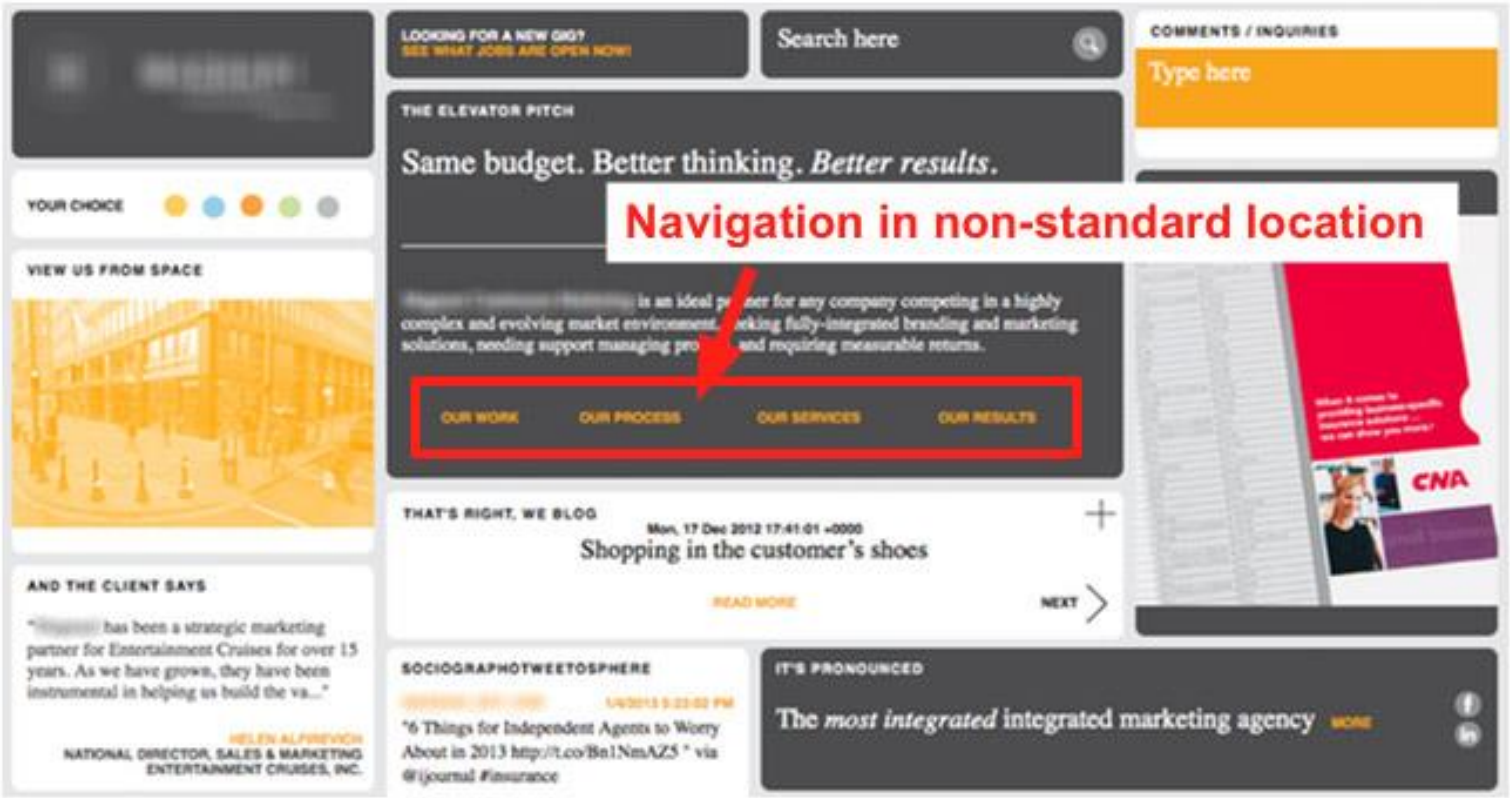
- Courtesy navigation
  - Provides access to items that users don't need on a regular basis, but that are commonly provided as a convenience



  - Example: links to contact information, feedback forms, and policy statements
- Remote navigation tools
  - Navigational devices aren't embedded within the structure of the pages, but stand on their own
  - Users turn to them when they get frustrated with the other navigational systems
  - Site map: gives users a concise, one-page snapshot of the overall site architecture
  - Index: alphabetical list of topics with links to relevant pages

Common navigation mistakes

1. Non-standard navigation style



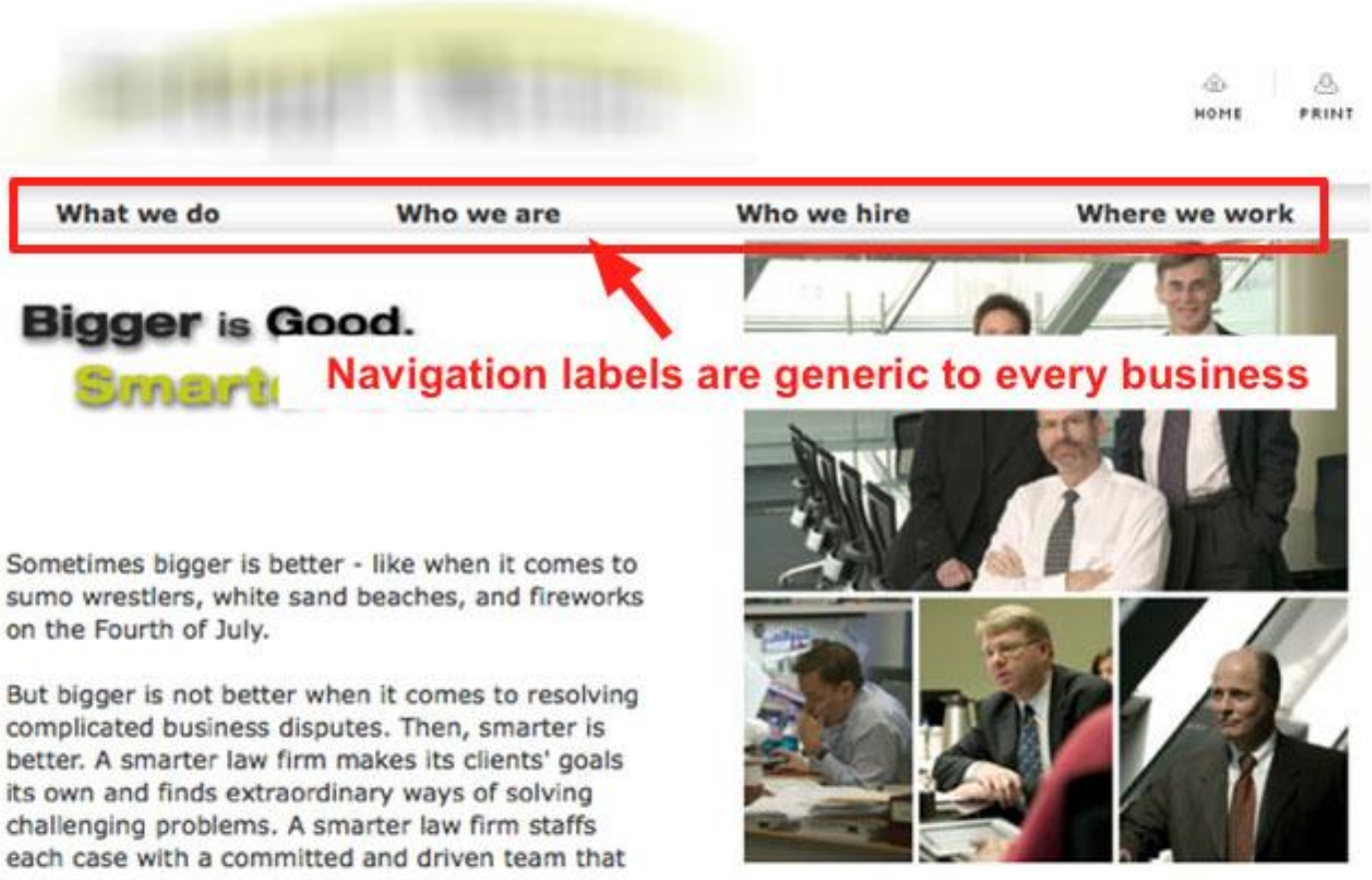
https://blog.kissmetrics.com

Common navigation mistakes

1. Non-standard navigation style
   - Putting navigation in standard places makes the site easier to use
   - Visitors expect to find horizontal navigation across the top or vertical navigation down the left side
   - Marketing is about differentiation, but navigation style isn't the right place

Common navigation mistakes

2. Use generic labels



Common navigation mistakes

2. Use generic labels
   - Navigation should be descriptive
   - Labels like “Products” or “Services” are generic to all businesses and do nothing to communicate with visitors
   - “What we do” doesn’t say what you do
   - When navigation shows your main services or products, your site will communicate instantly
   - The navigation throughout the site and the site’s structure itself should be planned with search engines in mind
   - Navigation is also a huge opportunity to indicate your relevance to search engines, and your audience isn’t searching for “products” or “services”

Common navigation mistakes

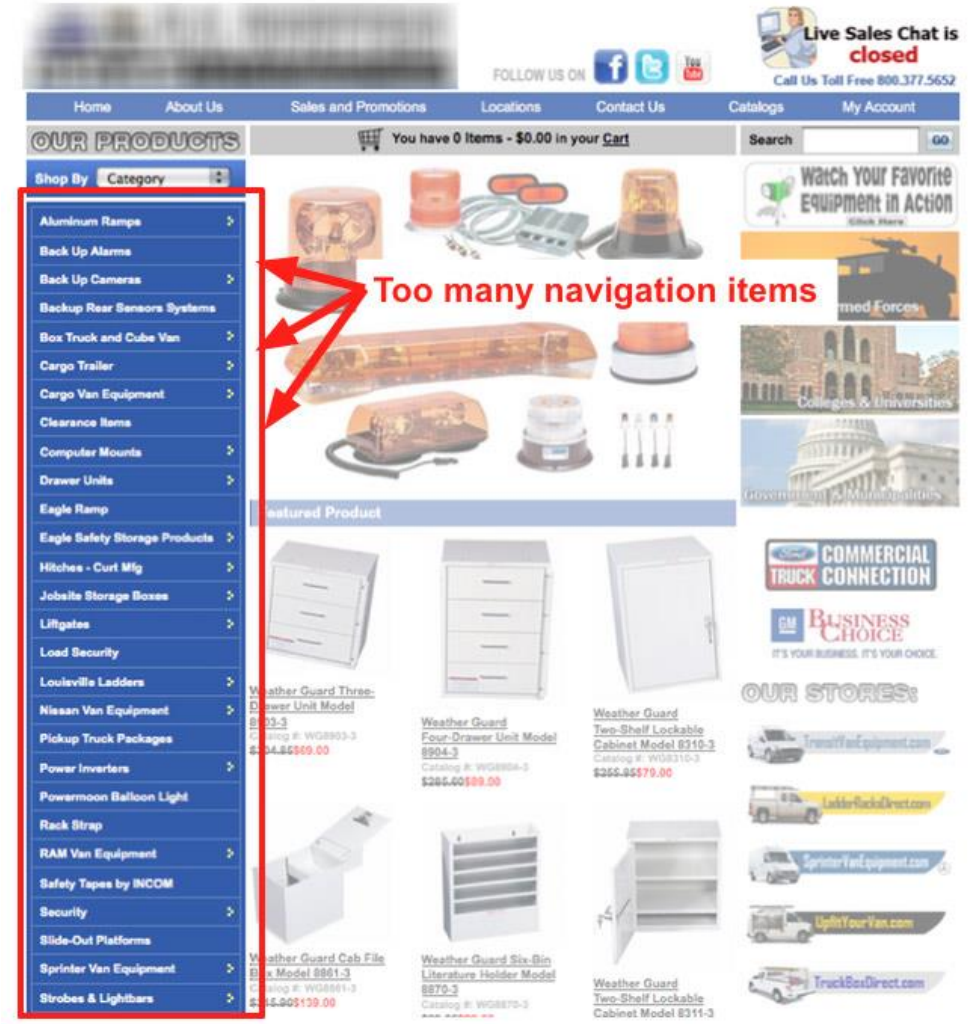
3. Drop down menus

Common navigation mistakes

3. Drop down menus
   - Depending on how they're programmed, can be difficult for search engines to crawl
   - Annoying, according to usability studies (Nielsen-Norman Group), because visitors move their eyes much faster than they move the mouse; when they move the mouse to a menu item, they have already decided to click…and then the drop down gives more options
   - Encourage visitors to skip important top-level pages
   - Exception: really big "mega drop downs" with lots of options test well in usability studies

Common navigation mistakes

4. Too many items

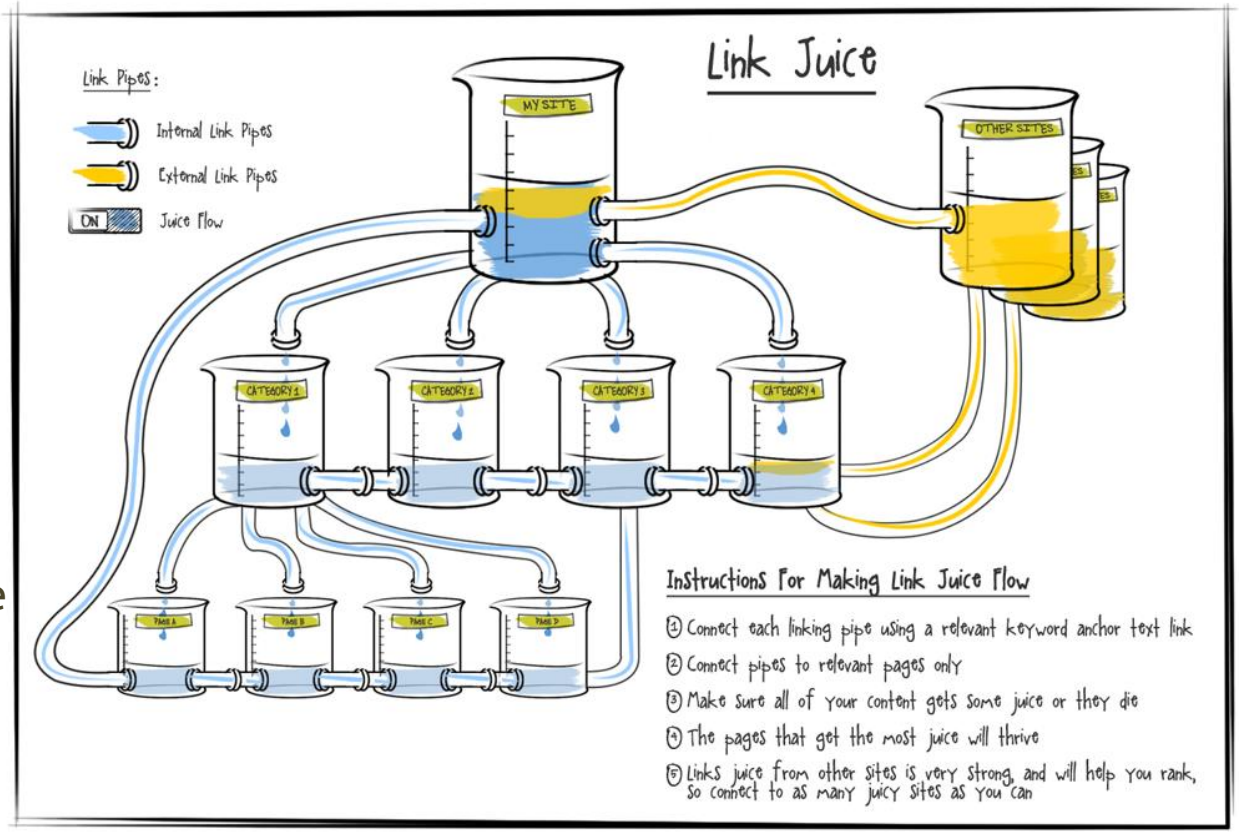


Common navigation mistakes

4. Too many items
   - Websites with hundreds of links on the home page are terrible
   - Even eight links may be too many: short term memory holds only seven items
   - With fewer menu items, visitors' eyes are less likely to miss important items
   - Every time you remove a menu item, the remaining items become more prominent
   - This also works for the rest of the page: every visual element removed makes everything remaining more prominent
   - Concise navigation is also important for search engine optimization: when your navigation has too many links, less authority and trust passes down to the interior pages: the link juice is diluted
- "Link juice" is a term used to explain the relative power that any link can give to another webpage
  - Links can send link juice to another page on the same website, or to an external page on a different website
  - Each link has a certain amount of link Juice, and some links have more powerful link juice than others

Link juice

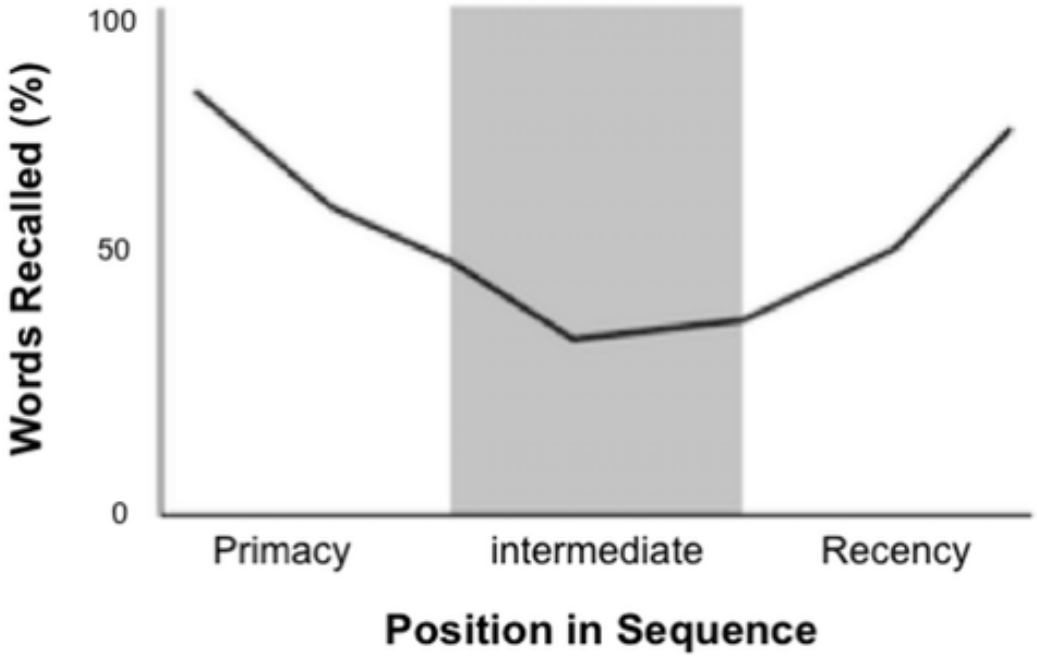
- There are a number of factors that affect the power of a link's link juice, including the power of the referring webpage and the number of other links that are on that page
  - For example, a link from a powerful page will have more "link juice" than a link from a weaker page, and a link from a page with many other links on it will have less "link juice" than a link from a page with only one link on it



https://www.linkdex.com/

Common navigation mistakes

5. Getting the order wrong



Common navigation mistakes

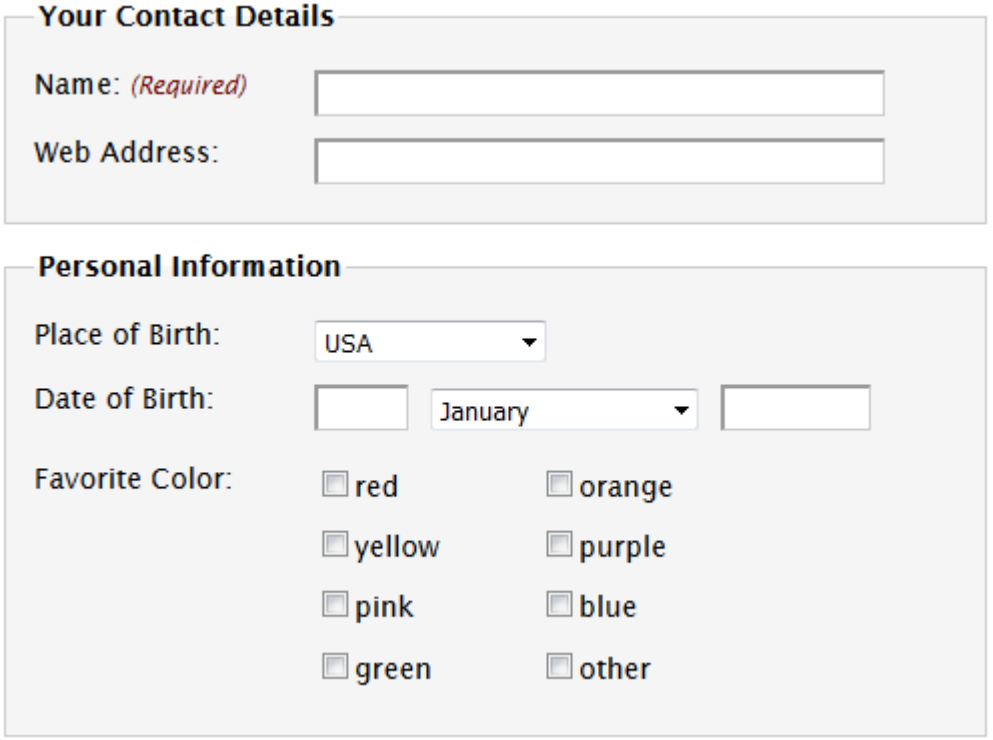
5. Getting the order wrong
   - Items that appear first or last on any list are most effective, and navigation is not an exception
   - Psychology studies show that, attention and retention are highest for things that appear at the beginning and at the end: it's called the "serial position effect," and it's based on the principles of primacy and recency
   - Put your most important items at the beginning of the navigation and the least important items in the middle
   - "Contact" should be the last item on the list, putting it at the far right in top-level horizontal navigation, a standard location

Common navigation mistakes

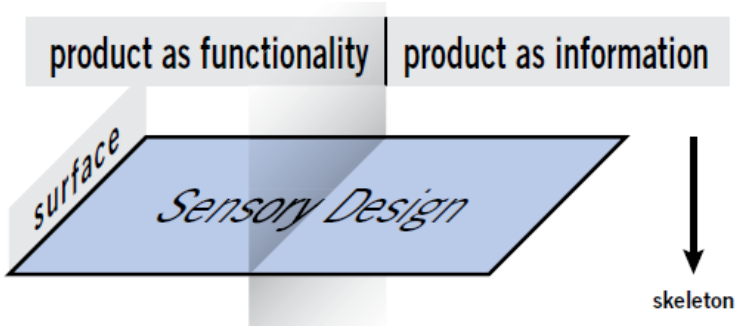
6. Use buttons instead of links
   - Buttons are not search friendly, since the text within is invisible to search engines
   - Buttons are harder to update than links, requiring Photoshop and a new image for every update
   - Buttons load more slowly that links, making them especially bad for mobile visitors
   - Buttons are less accessible to the visually impaired
   - Buttons are unnecessary, even if you want to use non-standard fonts (e.g. use tools like TypeKit)

Information design

- Making decisions about how to present information so that people can use it or understand it more easily
- Sometimes is visual
  - Pie charts or bar charts?
- Sometimes involves grouping or arranging pieces of information



The surface plane



- Sensory design
  - The aspects of the product the users will notice first
  - The term extends «visual design»
- Content, functionality and aesthetics come together to produce a finished design that pleases the senses while fulfilling all the goals of the other four planes
- Visual treatment of text, graphic page element, use of sound and navigational components
- Graphic and sound treatment of interface element (the «look» in «look and feel»)
- Effective tools for attracting and directing attention
  - Empty space, contrast, colors, …

Follow the eye

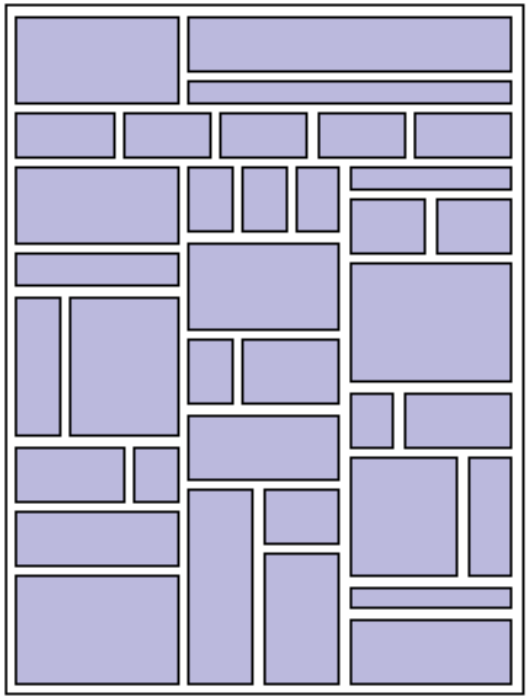
- The movement of the user's eyes around the page doesn't happen by accident
  - It's the result of a complex set of deeply instinctive responses to visual stimuli that all humans share
- If your design is successful, the pattern the user's eye follows has two important qualities
  - It follows a smooth flow: when user comment that a design is "busy" or "cluttered", it means that their eyes bounce back and forth among a variety of elements all clamoring for their attention
  - It gives users a sort of guided tour of the available possibilities without overwhelming them with detail: possibilities shouldn't distract from information or functions that users will need to fulfill their goals

Contrast and uniformity

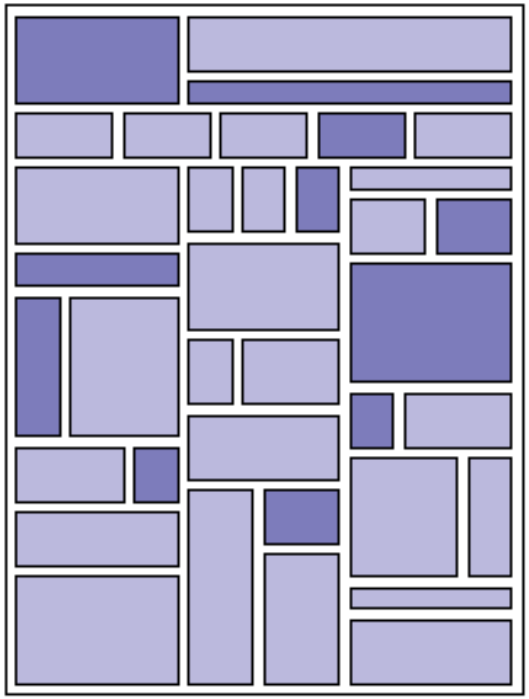
- Contrast
  - Is vital to drawing the user's attention to essential aspects of the interface
  - Helps the user understand the relationships between the navigational elements on the page
  - Is the primary means of communicating conceptual groups in information design
  - Example: error messages
- The contrast has to be significant enough for the user to clearly tell that the design choice is intended to communicate something
- Maintaining uniformity in design is an important part of ensuring that your design communicates effectively without confusing or overwhelming your users
  - E.g., keeping the sizes of elements uniform can make it easier to recombine them into new designs as you need them

Contrast example

- Neutral layout (no contrast)

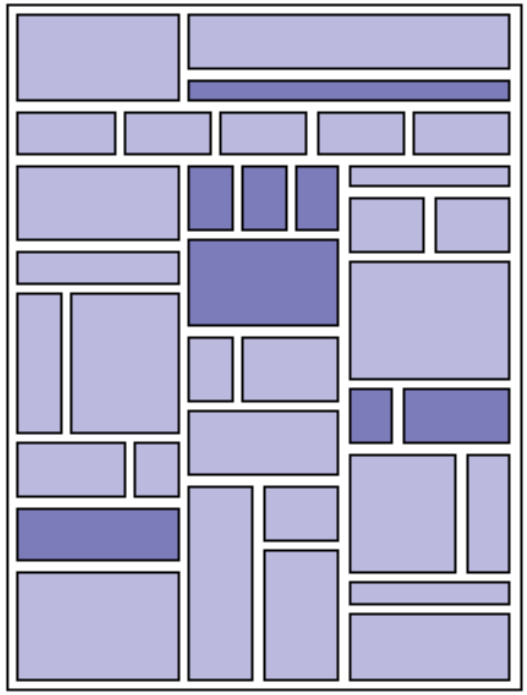


- Contrast used to guide user's eyes around the page

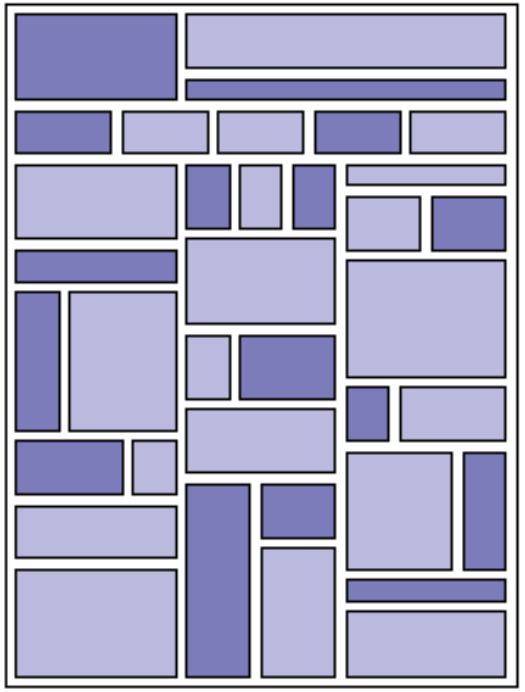


Contrast example

- Contrast used to draw attention to a few key elements

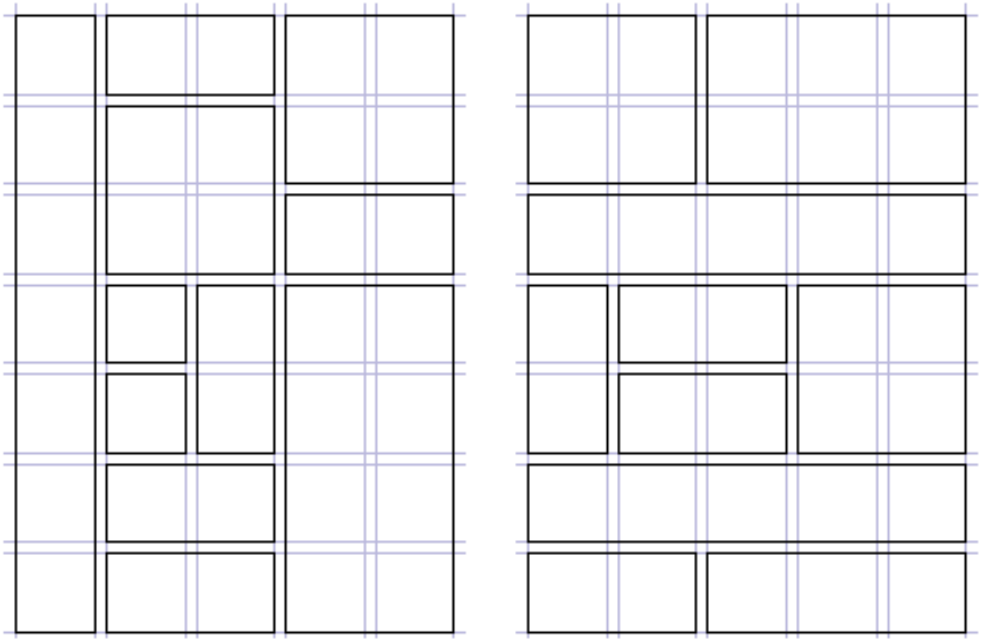


- Overuse of contrast



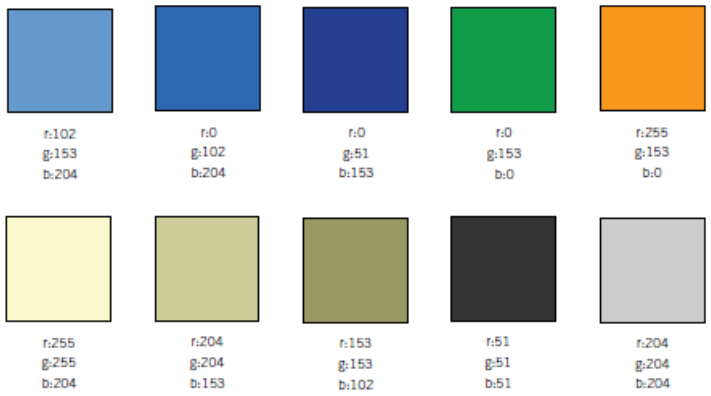
Grid-based layout

- Ensures uniformity of design through a master layout that is used as a template for creating layout variations



Color palette and typography

- The core brand colors are usually part of a broader color palette used in all of a company's materials
- For body text, or any material that will be presented in larger blocks, the simpler the better
- For larger text elements or short labels like those on navigational elements, typefaces with a little more personality are appropriate
- Few fonts, not to overwhelm users



Icons

- Benefits of icons
  - Make good targets: they are typically sized large enough to be easily touched, but also work well with a mouse cursor
  - Save space: can be compact enough to allow toolbars, palettes, and so on to display many icons in a relatively small space
  - Are fast to recognize at a glance (if well designed)
  - There is no need to translate icons for international users, provided that the icons are mindful of cultural differences (e.g., mailboxes look very different in various countries whereas envelopes look the same)
  - Can be visually pleasing and enhance the aesthetic appeal of a design
  - Support the notion of a product family or suite when the same icons and style are used in several places

Icons

- Downsides
  - “Universal” icons are rare (e.g., home, print, search)
  - Icons often need a text label
  - The relative size of an icon matters: mobile app vs. desktop application

- The 5-second rule: if it takes you more than 5 seconds to think of an appropriate icon for something, it is unlikely that an icon can effectively communicate that meaning

Icons

- Icons for website navigation: are they worth?
  - When well-designed they look nice, but if you removed from a site it would be none the worse for it…
  - …probably the cost to develop them is not worth it
- Valuable design time would most likely be better spent elsewhere

References

- J. J. Garett, “The Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition”, Pearson
- J. Porter, “Designing for the Social”, New Riders
- A. Dix, J. Finlay, G. D. Abouw, R. Beale, “Human-Computer Interaction, 3rd Edition”, Pearson

License

- This work is licensed under the Creative Commons “Attribution-NonCommercial-ShareAlike Unported (CC BY-NC-SA 3,0)” License.
- You are free:
  - to Share - to copy, distribute and transmit the work
  - to Remix - to adapt the work
- Under the following conditions:
  - Attribution - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
  - Noncommercial - You may not use this work for commercial purposes.
  - Share Alike - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.
- To view a copy of this license, visit http://creativecommons.org/license/by-nc-sa/3.0/