Notes on the Software Design Process

» UCD phases & artifacts
» Organizational models
» Leadership issues
» Information architecture
UI Designer at Involution Studios
Private design studio in Santa Clara
- Previously: frogdesign, Adobe, BEA, Oracle
- Master’s in Interaction Design from Carnegie Mellon

Lecture Overview
- Role/place/function of design in organizations
- Scope of design activity within software development
- UCD process phases and artifacts
- Issues of design leadership
- What is information architecture
Evolution of the Software Development Process

1. Originally, programmers did it all:
   In the early days of the software industry, smart programmers dreamed up useful software, wrote it, and even tested it on their own. But as their businesses grew, the software business and software products became more complicated.

2. Managers brought order:
   Inevitably, professional managers were brought in. Good product managers understand the market and competitors. They define software products by creating requirements documents. Often, however, requirements are little more than a list of features, and managers find themselves having to give up features in order to meet schedules.

3. Testing and design became separate steps:
   As the industry matured, testing became a separate discipline and a separate step in the process. In the move from command-line to graphical user interface, design and usability also became involved in the process, though often only at the end, and often only affecting visual presentation. Today, common practice includes simultaneous coding and design followed by bug and user testing and then revision.

4. Design must precede the programming effort:
   A goal-directed approach to software development means that all decisions proceed from a formal definition of the user and his or her goals. Definition of the user and user goals is the responsibility of the designers—thus design must precede programming.

Figure 1-1: The evolution of the software development process. Today, design is often an afterthought. It should, instead, happen before any coding or testing begins.

~ Alan Cooper, About Face 2.0
Rational Unified Process

Iterative software development process framework
Based on a set of key principles for business-driven development

1. Adapt the process
2. Balance stakeholder priorities
3. Collaborate across teams
4. Demonstrate value iteratively
5. Elevate the level of abstraction
6. Focus continuously on quality

Agile Programming

Most agile methods attempt to minimize risk by developing software in short iterations, which typically last one to four weeks. Each iteration is like a miniature software project; reaction against waterfall method.

Steps: planning, requirements analysis, design, coding, testing, and documentation

Extreme Programming

XP regards ongoing changes to requirements as a natural, inescapable and desirable aspect of software development projects; they believe that being able to adapt to changing requirements at any point during the project life is a more realistic and better approach.
Figure 6.2 Overlap of disciplines leads to value: User-centered iNPD.

~ Cagan/Vogel, Creating Breakthrough Products

iNPD Model Overview
UCD Process

Understand *exploratory*

User goals, values, motives, desires

Create *generative*

Model *interpretive*
UCD Artifacts/Phases

Exploratory
- Scenarios
- Personas
- User Roles
- Role/Task Matrix

Interpretive
- Object Models
- Info Architectures
- Taskflow Diagrams

User goals, values, motives, desires

Generative
- Wireframes
- Mock-ups
- Prototypes
UCD Artifacts/Phases

Interpretive
- Object Models
- Info Architectures
- Taskflow Diagrams

Exploratory
- Scenarios
- Personas
- User Roles
- Role/Task Matrix

User goals, values, motives, desires

Generative
- Wireframes
- Mock-ups
- Prototypes

UCD Artifacts/Phases
UCD Artifacts/Phases

**Interpretive**
- Object Models
- Info Architectures
- Taskflow Diagrams

**Generative**
- Wireframes
- Mock-ups
- Prototypes

**Exploratory**
- Scenarios
- Personas
- User Roles
- Role/Task Matrix

User goals, values, motives, desires
Generative
Centralized enforcement group

Advanced concepts “splinter cell”

Embedded within engineering team

Centrally managed, BU-funded designers

Internal consultancy model

External pay-for-hire agency
Adaptive humanist leader

Be like ecologists: system of consequences

Total product lifecycle view of situation

Ask critical questions

Persuasive communication/influence

Hidden dependencies/support group

Designing anything involves satisfying constraints, making choices, containing costs, and accepting compromises.

Henry Petroski
What is IA?

The art and science of structuring and organizing information environments to help people achieve their goals.

Lou Rosenfeld, Argus Assoc.

IA encompasses all the design and structure from the back-end to through the content to the resulting representation necessary to create an information system useful to end-users.

Vivian Bliss, MS Library

Information architecture is a systematic, question-based process for creating communication products that help users meet their needs.

Thom Haller, Info.design

IA is the practice of creating plans that describe the underlying organizational structure for a system of content and interactions.

Steven Ritchey, Sapient

Application IA: Oracle Financials App

Intranet IA: Web-based CMS & Style Guide
We are searching for some kind of harmony between two intangibles: a form which we have not yet designed and a context which we cannot properly describe.

-- Christopher Alexander

Our guiding principle is that design is neither an intellectual nor a material affair, but simply an integral part of the stuff of life, necessary for everyone in a civilized society.

-- Walter Gropius