**Introduction:**

**Business, SA, SDLC, Modeling**

**INLS 582\_001, Systems Analysis**

**Monday, 1/9/12, Wednesday, 1/11/12, Wednesday, 1/18/12, Monday 1/23/12**

NOTE: no class Monday, 1/16/12, MLK

**Reading:**

1/11/12 Sellen & Harper, (2002), Ch. 7

1/18/12 Beyer & Holtzblatt (1998), Ch. 1

 Jaferian, Botta, Hawkey, Beznosov (2009) identity management case study

1/23/12 Hendry, D. (2004), Communication Functions…

**Assignment:** Case Study discussions. Discuss Monday 1/19/12, ask for 7 Volunteers for Wednesday, 1/18/12.

Rest should send me preference for CS assignment no later than Monday 1/30/12.

Discuss project, Wednesday 1/11/12, examples.

**Slides:**

sa-overview.ppt

sdlc.ppt

model-exercise.ppt

**Monday 1/23/12**, 9/1/11 create a model of SILS

**Discuss** expectations of group partitipation in small group exercises.

# Business

Introduction of me.

Introductions of them. Experience with large projects (development, planning, etc.)? What do you want to be when you grow up?

What’s the largest project you’ve ever planned?

Overview of course, web materials, project, etc.

<sa-overview.ppt>

List: Subscribe to list: go to mail.unc.edu/lists/

List name is inls582\_001

Sakai Site: Dropbox for submitting assignments

Web page. URL <http://ils.unc.edu/~stephani/sasp12/Index.html>

 For password-protected pages, the user name is sasp12, the password is ci3#views

Laptop policy

Go through website,

**Case Study:volunteers, rationale, roles**

Variety of problems, issues, locations.

Expand on what we can do in class

Look for themes, problems, approaches that could occur in any setting

What about a setting/context makes a difference? What might it look like in your chosen context/setting?

What trends, ideas, technologies, issues should you follow as a prefessional, as a person, in your other roles?

**Myth Themes**

* information ecology
* affordance
* automation boundary
* affordance
* community of practice
* organization boundary/system boundary>

**Sellen & Harper, 2002** I highly recommend their book “Myth of the Paperless Office”, especially the case studies.

This chapter has a lot to say about **goals of system design and redesign in general**, not just as it applies to decisions to move from a paper-based system to an electronic one. Although many projects we work on focus on possibilities of replacing or modifying paper systems.

p. 185 “the role of paper will continue to evolve and change”. This is true of any artifact or technology we use. As designers, we need to be sensitive to the change process.

**Affordances**

p. 186. 3 reasons that people stick with paper:

1. Coevolution of paper and work practices. The way we do things is designed to work with paper.

2. need for better design of digital alternatives. If the digital replacement is awkward, risky, inconvenient (or perceived to be that way), then it isn’t going to be a success. Compare layering 3 paper forms on your desk to 3 electronic forms on the screen.

3. affordances of paper. What are the characteristics of paper that allow or promote kinds of work? E.g., folding, annotation, portability, sharing.

p. 187 2 themes in the book that I generalize here:

1. Change in work tools or processes can often assume a symbolic role. “out with the old, in with the new”, progress, old or traditional is bad, shift in power, etc. We’ll talk about these throughout the term.

2. artifacts and processes evolve together. Changing one usually requires changing the other.

p. 188 What is an office?

“In various ways, all this research has shown that effective offices of the present day are not simply a function of the way an office environment is laid out or the various information tools and technologies in the office (whether they be paper or digital media). Nor are they simply a function of the people who populate an office or the work processes they engage in. They are a product of the *interaction* of these things.”

Elicit definitions of terms on board (see earlier)

**Norman, Donald (2007). The Design of Future Things. New York: Basic Books**

"Affordance… is really communication between the designer and user of a product" Norman quoting Clarisse de Souza. P. 66

In designing an artifact, including a system, affordances need to be obvious to the user, especially for infrequent or unfamiliar users/uses. <does "obvious" always mean "visible"?>

Signals and triggers are another important aspect of design. Norman talks about the cacaphony of beeps and lights that we are surrounded with (e.g., in car, microwave). Problems: overload, ignoring, differentiating "normal" from "action needed".

He describes the sound of a teakettle as a good example: from the burble as the water starts heating up, the very beginning of the steam coming out, and the whistle. It's connected wit the natural function, it clearly describes the progress, it's builds in urgency.

Back to Sellon…

p. 191, they discuss the vision of the future office as an “information window” (laptop) and a “communication tool”(PDA). What do you think of this? Workable? Likely? Attractive? Are we there yet? (RENCI immersion room)

p. 193 lessons in system design

* focus on the real underlying problems
* be willing to revise visions, reassess solutions
* manage expectations

In any project, you need to consider when t focus on the present and when to focus on the future.

* When should you base design ideas on existing practice, functions, and needs?
* When should you try to break loose completely from current ways of working?

p. 203 the document life cycle. Many projects deal with some aspect of this – consider it in your project if applicable.

What can we pull from this chapter for 582? The importance of finding an appropriate solution, not just a cool one. Solution(s) must take into account:

* Organizational culture
* Organization’s resources
* Nature of the change you propose
* Short/long term change needed
* High/low end solution possible.

**Discuss project, project ideas**

# What is an information system, and why analyze it?

Elicit definitions of system, information system.

American National Standards Committee: “In data processing, a collection of men, machines, and methods organized to accomplish a set of specific functions.”

Includes environment in which task is done, interface with outside of system. Systems interact with other systems.

Alter (2001) uses the term **work system**. “a work system is a system in which human participants and/or machines perform a business process using information, technology, and other resources to produce products and/or services for internal or external customers” (p. 7)

He defines an **information system** as a type of work system: “Information systems are a special case of work system in which the business process is devoted to only six types of activities: capturing, transmitting, storing, retrieving, manipulating, and displaying data….The purpose of most information systems is to support one or more work systems.” (p. 8)

Davis: “A **system** is a set of components that function together in a meaningful way.” (p. 6).

“An information system is a set of hardware, software, data, procedural, and human components that work together to generate, collect, store, retrieve, process, analyze and/or distribute information. The purpose of an information system is to get the right information to the right people at the right time.” (p. 7)

<capture what data? Do what with it? Disseminate what to whom? Store what?>

<Are there any information systems that don't fit this definition?

Note: I add "…in the right form">

**General Principles of Systems: <SDLC.ppt, slide 1>**

1. Systems are part of larger systems. Any system can be partitioned into subsystems. (Functional decomposition). The boundary of any system could be a little bigger or smaller. Example: We can look at Land’s end warehouse picking system, or expand to look at the order/warehouse system, or the order/warehouse/delivery system, and so on. Another example is registration system, which interacts with cashier, health, advisors.

It’s important to consider the context or environment in which a system exists.

2. Systems grow. In amount of data, number of users, number of transactions, kinds of tasks and functions, geographical area of coverage, etc. One part of analysis and design is to try to plan for the kinds of growth that will occur. This is hard, because of technical innovations, organizational changes (merging companies, etc.)

3. The larger a system, the more time is spent in housekeeping. By “housekeeping”, I mean the kinds of organizational tasks that aren’t directly in the workflow of producing the information product. Communications, confirmation, sorting, printing, fixing errors, etc. For example, if you have only F2F reference, then prioritizing who serves whom is easy. As you add more services, e.g., email, phone, chat, FAQs, pathfinders, etc., prioritizing serving, updating, etc. becomes more complex. More time must be spent on scheduling, allocating time, switching between modes.

4. Changing a system, or part of one, is difficult, and takes planning.

# Systems Analysis is Problem Solving

A problem occurs when you are currently in one state or situation, and you would like to be in a different state or situation. Some actions or changes must be made, but it isn’t entirely clear how to move from the current to the desired state.

Broadly stated, the problem is that the current system (or lack of one) isn’t producing the information that people need when and how they need it. One or more functions of the system are breaking down, or just don’t exist. But we need more detail to figure out the specifics of the problem, and what will help us solve it. This is where systems analysis comes in. Procedures, methods, tools, models, to help identify the problem(s), and design a new system that will solve it.

How can you tell when there is a problem with an information system?

* Easy case: The organization is starting something new, needs new tasks/functions, and there is nothing in place. Designing the system from scratch.

Other situations aren’t as clear.

* Symptoms that the system isn’t performing as well as it could/should.
* Speed is too slow.
* Can’t handle transaction load
* Too many errors
* High employee turnover (unpleasant job)
* External forces require change – legislation, competition, coordination.
* Changes in the organization itself require change – reorganization, relocation, take-overs.
* User dissatisfaction

Symptoms may show up gradually – **graceful degradation** – or be more sudden, even catastrophic.

Reverse engineering, re-engineering, business process re-engineering, brp – these are all terms you may hear.

# Goals of Systems Analysis

Need to determine both symptoms of the problem and what is causing them (the underlying problem) before you can solve it.

Need to determine functions, interface, work flow, data, etc. of new system, and determine what needs to be kept or modified of existing system.

Need to plan the project – what resources do you need? What people? Who do you need to talk to? Where are the risks? A project may be minor, taking only a couple of months and affecting only a couple of people, or major, taking years and affecting thousands of users, customers both directly and indirectly

Need to get good time/cost estimates for the project (a notorious problem for systems and other projects – consider the RDU parking garage).

Much of systems analysis and design is concerned with gathering and organizing information. The methods and organizational strategies are done using a variety of tools and models.

A goal of this class is to give you a variety of methods and tools, and help you learn when each is useful.

**Case Study**

J[aferian, P., Botta, D., Hawkey, K., & Beznosov, B. (2009). A case study of enterprise identity management system adoption in an insurance organization.](http://libproxy.lib.unc.edu/login?url=http://doi.acm.org/10.1145/1641587.1641594) Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology (CHIMIT), 46-55.

Identity management (IdM) creation/maintenance of user's digital identities, access to resources, ganting authority, accountability, compliance. <think about identity management just for this class – for your life as a student, employee, person, etc.>

Entity IdM within the scope of organization, roles, etc.

High risk: legal, monetary, social

Need for case studies: what works, what doesn't, what research is needed.

Insurance organization.

4 semi-structured interviews from Security Administration involved in IdM syste deployment. <how do you identify good people to interview? depends on what you need to learn>

Different stages of IdM adoption process: IdM selection, deployment Phase1, Deployment Phase II

Analysis: direct interpretation, categorical aggregation. – deconstruction, contextualizing, merging to look at full picture.

High-level coding: stakeholders, tools, challenges, tasks, interactions.

also open-coding (definitions, if needed)

Phase 1: self-service password recovery; basic provisioning of services, access to all employees..

centralized management of digital identies; distributed access control.

Policy development: who can access a resource, who "owns" the resource (data guardian). guardian grants access, but can designate a more knowledgable data steward if desired.

Problem: situation before IdM System.

Process/workflow. (p 48). combination of automatic and manual steps, multiple actors.(every guardian/steward of a needed resource). Includes a follow-up step in case of delays. If employee terminated, manager should notify SA group.

Challenges. Timelag between employment and access. reptition of action. Impossible forms needed; requesting hard information. Over-broad requests for access. Significant knowledge needed by manager, including terminology/concepts. Keeping track of access priveleges was hard.

<note use of quotations to illustrate points>

Employees tend to accumulate privelges as role/job changed, but never lose unneeded ones. No penalty if manager didn't alert SA about change or termination. Work-around: SA queries DB for accounts not used for a while, follow-up manually.

Difficult to perform audits, troubleshoot. Poor record of accesses.

Sloppiness with passwords, sharing. Not following policy.

Deployment of IdM.

Need to get policies, procedures set up before deployment. Need to clean up roles, rules, standardize workflow and definitions.

Need to involve/educate stakeholders. < are stakeholders only direct users/employees?>

Incremental deployment, to ease transition. First change was one that people would really notice, and appreciate (self service passwords). That smoothes the way for subsequent changes.

Analysis of staff needed for deployment; esp. knowledge (security business analyst)

Note difficulty in getting estemate from vendors, given that vendor couldn't see actual current system, needs. So they hired a consultant to prepare a descriptive report to give to vendors, and they based estimates on that.

Requesting bids: requirements; establishing scoring system for bids. then get presentation from best vendors. Then they did lab tests on 2 best <note that this is large company, with resources, for whom security and IdM is crucial. time/money must be spent, and vendors understand the need. IOW, the company can make demands of competing vendors.

NOote the problem of time: vendor claimed 3 months, company allowed 1 year, it actually took 15 months.

Note how vendor and organization become partners, each benefitting in other ways (experience, education, references)

Expected benefits: automatic provisioning of new employees, based on roles. reduced workload. Online forms. Better reporting/compliance.

challenges: hard to define/engineer roles, even using role mining. need for collaboration with business side of org.

Getting all components of system to work dogether was hard.

Difficult to plan/rehearse deployment in operational environment.

Lessons learned:

importance of defining guidelines/criteria to create requirements.

e.g., integration with current infrastructure, workflow support, good reporting/presentation of system information, both for audits and to demonstrate importance/contribution of system and department to company.

UX, archiving

need for planning/rehearsal.

need for domain expert.

# Systems Development Life Cycles

Elicit what SDLC is, why needed, in small groups:

What are components of a successful development project? (tasks, deliverables, etc.)

Each group create their own list, compare/contrast.

# Purposes of any Life Cycle Model

The idea is to organize the tasks, and recognize that systems are born, implemented, undergo modifications, and finally, must either have major re-design (re-engineering), or be scrapped. They get too brittle.

1. Define the activities that must be carried out. Don’t leave out a crucial step. This also makes working in a team easier.

2. Provide a consistent approach in development projects. Allow experience to count and make things easier.

3. Provide checkpoints for where you are in the project, what you should be doing, what decisions should be made. For example, at several points in the waterfall mode, you have an opportunity to decide whether to continue the project. At various stages, management may want the option to decide, “given what we know now, should the project continue?” There should be clear information provided to make these decisions. The decision gets more expensive to make as the project progresses, in terms of what has already been spent; an organized approach helps assure that the project hasn’t just been spinning wheels. Note that this is especially important in large systems that may take many months or years.

4. A model is a way of communicating with other team members and your customers, to help you understand what you will be doing and when. This is especially important when working with people with different experience and background (which happens much of the time). It is important to manage expectations!

When we read Brooks (1995) in the project management section, you may also get the idea that another reason for having a plan is so you can tell when you’re in trouble!

Textbooks and “method” books present lots of variations of development plans.

**Traditional Waterfall Style Life Cycles: (SDLC.ppt, slide 3)**

In a waterfall style, the project is divided into distinct phases. Each phase is completed before the next one starts. Each phase generally has its own goals, cost estimates, time estimates, resource estimates, products (e.g., design documents), and is intended to lead to more precise estimates on the success and cost of the next phase and the rest of the life cycle. A formal decision to continue or not to continue may be made at the conclusion of each phase.

1. **Problem Definition:** *What is the problem?* Results from a user complaint, problem with performance, etc. Results in written statement of what the problem is, what the goals and objectives are in terms of solving it, what the scope of the problem is and what the scope of the investigation will be (and therefore, what may be considered fair game for change). Also results in general estimate of cost of project. This may take less than a day to do. If this is approved, then the next step, involving slightly more time (money) is carried out, otherwise the decision is made to live with the problem.

2. **Feasibility study**: *Is there a feasible solution to the problem?* What does a feasible solution look like? Recall Clemons, with kinds of risk. More specifically, what are the goals of the project? What pieces will NOT be considered (are outside of the scope of the project). Results in more specific goals and objectives, and a somewhat more precise cost/benefit analysis. These are presented to decision makers, who decide whether to proceed with the project.

3. **Analysis:** *What must be done to solve the problem?* Develop a logical model of the system (dataflow diagrams, data dictionary, E-R model, object model) that will provide a system to solve the problem. These should give the users and other interested parties enough information to be sure the analyst is on the right track, is looking at the right problem, is incorporating the important functions.

4. **System Design:** *How should be problem be solved?* Move from the logical model to the physical model. Consider alternative ways of realizing the model. Produce system specifications, including alternatives. (E.g., can produce a very basic solution, that just fulfills the requirements, a middle solution that may include other features, and a top-of-the-line solution.) Develop detailed specifications for each procedures, specific formats for data, determine hardware specifications

5. **Development:** *Do it.* Order and install hardware, write software, develop interfaces, write training manuals, operating procedures, security plans, auditing procedures.

6. **Testing:** *Does it work as it is supposed to?* You must both determine how to adequately test the system, and actually test it. You cannot unleash an untested system. This doesn’t guarantee that the system works perfectly, but it should catch the more egregious errors. When the system passes the tests, the customer should sign-off, and accept the system.

7. **Maintenance:** This keeps the system operating during use. Maybe some new bugs will be discovered. Someone may try to use the system a little differently, and it can be tweaked to accommodate this. Maintain documentation. Adjust for growth. At some point, the system can no longer be maintained, and the cycle starts over.

There are variations on this Waterfall, but the general outline is the same. Davis also gives a modified version, in which there are iterative loops at each stage, allowing for the incorporation of feedback.

Problems with Waterfall model: speed of development, focus on process and models and documents rather than end product, keeping up with change, and lack of flexibility.

These are problems with any large-scale development project. If the project will take a year to complete, and major changes occur during that year, then either the end product doesn’t solve the problem any more, or you have to make changes along the way. Consider parking at the airport – the garage was a great idea when it started, but with the advent of more airlines and flights, it didn’t do enough from the day it opened.

In thinking of your projects, you may either be looking at the initial creation of a system, or a system that is in the operation and maintenance phase, ready to start a new iteration.

# Structured Analysis and Contextual Design

We’ll be using a combination of approaches this term. Mainly, we’ll look at a user-centered model, Contextual Design (B&H). Points of emphasis:

People are a crucial part of any organization, and therefore of any information system. A successful system must incorporate them in a satisfying way. Definitions of tasks, work flow, and so on must be examined from the human (and humane) perspective, as well as from the system perspective (think about their example of printing the envelope).

The term “deskilling” refers to re-engineering work flow so that it is efficient from the time/motion perspective, but has no intellectual interest to the worker.

The environment and context of any task or system is crucial for designing a successful system. You can’t design in a vacuum. An important part of gathering information about the needs of the users and the system is to look at the context. Go to the place and watch someone do the task. Once you have a prototype, have someone actually try to use it.

B&H also talk about aspects of the analysis team and how it works. They give a variety of models, but also talk about how to build them. They acknowledge the presence of different perspectives on work, after an interview, and so on, and talk about integrating different perspectives, keeping the goals of the system in mind.

# Different parts of Contextual Design (p. 22-25) <SDLC.ppt, slide 4>

**Contextual Inquiry**: Understand customers, needs, desires, approach to work. Gathering information and coordinating it into coherent view.

**Work modeling**. What are the users doing, what are the constraints, requirements?

**Consolidation**. Putting all the components, different user populations, etc. together. Find the underlying patterns, rather than individuals’ idiosyncrasies.

**Work redesign**. Given what you know, invent improved ways of doing the work. What will the new system do, and how?

**User Environnent Design**. User interface, background processes, detailed components.

**Mock-up and test with customers**. They recommend quick-and-dirty paper prototypes for run-throughs, gathering feedback, making quick modifications.

**Putting into practice**. Development and deployment.

They also talk about managing change. This is crucial for adoption of a system – you are asking people to change. Even if they hate the old system, changing habits is hard. How do you prepare people? How do you communicate with them? Who do you need to communicate with? The lines of responsibility can be quite complex.

* Project team
* Project leader or coordinator
* Management
* Stakeholders
* Decision makers
* End user
* Indirect end users (e.g., you are an indirect end user of the motor vehicle department system – the clerk is the direct end user.)

Who is your client? Example – system for checking out physics demonstrations. The professors are one class of users – they’ll be doing the check out, and they’d like it as flexible as possible (they didn’t do any checkout before). The lab manager is the actual client, and he wants some tracking of who has what – he’d like some control).

Compare/contrast client and stakeholder

**Agile Approaches**

**Problem**:

* Traditional SDLC has a relatively long analysis/development time. Years for a large-scale project
* Requirements can change in that time, due to clarification or changes in need
* Environment changes in that time.
* Therefore, initial documents, plans are no longer relevant to organization's current needs.
* A commonly-held belief (backed up by some evidence) that changes made later in the development cycle are more costly and harder to do than changes made earlier.

What if we change the assumption from "changes are an exception, a problem, a sign of failure in requirement specifications" to "changes are normal".

The Agile development movement is about modifying SDLS to better accommodate changes.

Essentially, break down deliverables into small chunks with fast turnaround.

* less time for changes to creep in
* don't allow time for it to be "too late" to make changes
* Each deliverable is a final product.
* Cycle time is a matter of days or weeks, not months or years.

Frequent brief meetings (even daily) to check progress, problems. Never let any problem or concern go undetected by team.

Agile manifesto agilemanifesto.org

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Cusomer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more."

Several versions of this:

Agile Alliance [www.agilealliance.org](http://www.agilealliance.org)

Emphasis:

F2F teams, work in group setting, talent, skills, communication, individual drive for group to succeed. self-organizing teams.

HIGHLY engaged clients – providing feedback on each deliverable, re-prioritizing next deliverable, features, decisions. (How realistic is this demand?) Team member, or an appointed client surrogate who can speak for clients for "everyday purposes"

**Modeling**

**Gause, Donald & Weinberg, Gerald (1989) Exploring Requirements: Quality Before Design**, New York: Dorset House Publishing

 p. 8 "*The most important quality of a map is that everyone involved should be able to understand it."* [emphasis theirs]

p. 9"Maps of Requirements are not Requirements"

"When the map and the territory don't agree, believe the territory" [ from Swedish army Training]

**Hendry, 2004**

This article talks about models and similar design documents as supporting vital communication functions within a design team, or between a design team and outside stakeholders (e.g., end-users or managers).

If you consider design as an individual activity, communication via models may be needed to elicit feedback, or to fit individuals’ work in with others’.

But design as a group activity is a collaborative, social activity, where decisions and solutions must be negotiated within the group. Design representations (models) support the communication process.

Especially with cross-disciplinary teams, supporting and promoting communication is essential: team members have different expertise, represent different interests or stakeholders, or different departments. These are boundaries that must be communicated over: design representations can help.

<As we talk about our projects, what are the boundaries within teams? Between the team and the client and others (like me)?>

“Boundary objects are: ‘…objects which are both plastic enough to adapt to local needs of several parties employing them, yet robust enough to maintain a common identity across sites’” (p. 124). A useful model can serve as a boundary object, improving communication across boundaries.

He develops a list of 5 communication functions that design representations can serve, drawing on studies from the literature. “These studies suggest that representations are at least dual purpose. **First**, representations enable analytic and creative problem solving and the specification of solutions that can be judged against requirements. **Second,** representations enable particular kinds of communication among project stakeholders.” (p. 125, emphasis mine).

His list of 5 functions, from table 1, p. 125. <slide>

Conscripting – enlist participation and elicit reactions and comments

Coordinating – express progress and the provisionality of solution states

Framing – establish or reaffirm a common ground, typology, or constraint field

Persuading – convince a stakeholder, often a prospective client, that t solution fulfills project requirements

Recording – record the solution state to be used by others in the future.

Keep these functions in mind as you use each model this semester, and put together your final project specifications.

“Nevertheless, teams needed a method for making goals and tasks visible so that they could systematically study differences and identify areas for improvement.” (p. 129)

This explains a lot of what systems analysis is about: capturing and representing problems so there is a shared understanding and ability to propose alternatives.

**Bolloju & Leung (2006)**

This article describes common errors made by novice UML designers. Use ideas to help proofread your models – most of the points apply to any model, not just UML. Also consult my Guidelines document.

"Considering the fact that the reported failures of a significant percentage of developed systems are linked to faulty requirements, it is extremely important for these analysts and critical to the system's ultimate success to ensure the quality of the conceptual models they develop in the early phases of systems development." (p. 108)

**Types of errors (slide)**

Syntactic: Is the syntax used correctly? Components used to mean what they should. Good labeling. All required elements included.

Semantic: Does the model include everything necessary from the domain? Does it exclude unnecessary or irrelevant things? Does it capture things correctly? Is it complete and correct?

Pragmatic: Is it understandable by those who need to use it? Is it overly simple (and thus leaves out crucial information) or overly complex (so it is hard to understand)? Does it match the needs of the audience? Does it make sense?

**Exercise: model of SILS model-exercise.ppt>**

Prepare descriptions of model types that could be created for SILS/Manning. Hand one to each group.

Group must figure out model type, notation/components, what to include, what to exclude, what questions it lets you ask and answer.

 **[**

# Project Ideas

You need to come up with problems. Past projects:

* Web presence and bookkeeping streamlining for craftsperson
* Identifying and prioritizing communication and other organizational needs for a volunteer organization (NCSARDA)
* Redesign and re-implementation of IT database for large school
* Design and build database for highly distributed and unorganized information in Davis (consortia)
* Document management system for large school
* Organizing and tracking handout creation and storage for library
* Improving insurance enrollment process
* IT problem tracking for school
* Tracking/checkout system for physics demonstrations.
* System for recording data from series of experiments done by team of scientists under a variety of conditions, and producing reports for papers. Interesting because of high volume of data. Current system, everyone did it differently (spreadsheet, text file, etc.) so merging data for analysis was hard. Client thought DB was what was needed, turned out spreadsheet was best approach.
* Web catalog for videos at undergrad library, including availability.
* Tracking/circulation system for journals in a small department, including means of requesting copies of articles.
* Project management system for web development company. Hard project because of moving target – the company doubled in size during the course of the term.
* Membership information for village orchestra.
* Creation and tracking system for ruby accounts.
* Accounting system for small business
* Registration and tracking for ATN network
* Weeding and withdrawal for undergrad library
* Study abroad student orientation
* Placing orders in Davis library
* Arrival system for new faculty and staff at kenan-flagler
* Appointment scheduling at small medical practice
* Tracking reportable disease card
* Inventory and training at small bookstore
* Customer communication at small food business
* Gathering information from new hires at university school
* Gathering information to prepare for design of SOP tracking system at large business
* Redesign of conflict of interest form and process at university
* Redesign of bid system for small business
* Improving information dissemination for study abroad financial aid
* Redesign of web sites to identify user base; improve communication and image: exercise club, NSF center
* Weeding and maintenance plan for small special library folder collection
* Web architecture and deployment plan for wireless connectivity, for very large company
* Tracking changes to servers at server farm
* Coordinating e- and paper reserves at undergrad library
* Business plan for e-commerce and other marketing strategies for small business
* Knowledge management plan for university lab and help center
* Web search systems for library films
* Scheduling system for nurses in hospital department
* Reservation system for school computer equipment
* Reminder/checklist for faculty reviews
* Scheduling/education for department records retention
* Help desk task tracking
* Re-gifting of books donated to botany library
* Redesign of website for volunteer organization
* Design of e-commerce solution for small business.
* Document organization and maintenance policies on large library webpage
* Work flow in library preservation department
* Providing useful information to town IT department and users of town wireless
* Data analysis request workflow and tracking for major hospital analysis department
* Volunteer hour tracking and alumni tracking for student volunteer clinic
* Telecom change request system for major academic library
* Publicity release and news article tracking and archiving system for university department
* Alumni database for summer school
* Redesign interface for dental EPR at dental school
* Redesign of Blackboard site for public health online course
* Metadata contribution encouragement for ibiblio
* Tracking system for archives pre-acquisition
* Proposal components for small business
* PDA training for medical students
* Inventory maintenance for small business
* Knowledge management for student organizations
* Document flow for not-for-profit organization (ARC)
* Introducing a circulation system for small special library
* Organizing information on scholarship opportunities for university department
* Chemical inventory procedures for labs, in preparation for Homeland Security
* Small business work procedures to offload routine communications to webpage
* Data collection sheet for medical research studies for disease center
* Annotation practices for seminar
* Licence tracking for IT lab (with some work delegation)
* Requirements analysis for bus scheduling software
* Communication strategies for virtual research group with limited technical expertise
* Tracking contacts for DL acquisitions for university department
* Web presence for small business
* Student athlete training data
* Redesigining catalog and circulation for small film collection on campus
* Processing returns in a bricks-&-morter shop
* Coordinating budgeting data for projects in a biotech company
* Managing the music library at a small radio station
* Information management for IT department in university library
* Information tasks used by personal coaches
* Streamline quarterly report generation – merging information from multiple applications
* Training and workflow for concert venue employees
* Online history gallery for university radio station
* training and policies for public library volunteers
* Wiki for volunteer reviews of portal-provided health webpages
* Access plans for folk music DL
* Resource tracking for small department
* Training/information needs for small computer lab
* Work flow re-organization for animal fostering volunteer organization
* Website re-organization for community organization
* Design of weeding process for high school library
* Collection management plan for shelter library
* Knowledge/content management for business
* organization information on volunteers for international non-profit
* tracking work assignments for bibliographic instruction sessions
* redesigning a scholarly website for easy update (content management)
* evaluation tools for School IT service
* policy & procedures for handling confidential information in an archive
* policy & procedures for use of public computers in a library
* content and contact management for internship information
* inventory management for small used bookstore; moving away from paper
* policy & procedures for social media for a nonprofit
* web site design/content management for a small business
* library policies and materials for a church library
* tracking newly-purchased books for a public library
* information and web architecture for the Carolina StoryLab
* Data ingest workflow for data storage facility (Odum)
* System for requesting photos from archive
* database to manage bowling league data and statistics
* Process for cataloging archival collections for special library
* Communications organization (website, newsletter, vendor contacts) for farmers market
* In Agile-like environment, capturing ideas in organizational memory, rather than individual memory (library)
* Recommending/documenting process for creating exhibitions for very small special library with few resources (Stone)
* Scheduling/educating volunteers (CSW)
* Checking credentials on clinic volunteers (SHAC)
* improving sales/inventory tracking at small restaurant