Structured Application Development

• An Example of Application Development
• Why use a structured approach?
• One structured approach
  – Requirement Study
  – Prototyping
  – Implementation
  – Structured Testing
Application Development: An Example

• The National Center of Health Statistics has asked you to generate some tools to help them explore their mapped mortality data. They want to be able to view scatterplots that are linked to mapped data and animate through a time series.
• How should you approach this problem?
Why Use a Structured Approach?

• I can build a scatterplot!! Let’s do it!
• But do you know…
  – … who the intended users are?
  – … what the overall goals are for the application?
  – … where the data is coming from and in what units it will be mapped?
  – … what capabilities the scatterplot needs to have?
  – … how the will scatterplot work with the animation?
Why Use a Structured Approach?

• You need to answer these questions before you begin implementation!
• A structured approach …
  – … minimizes the number of stupid mistakes that you make.
  – … allows you to check back with your client on a regular basis to ensure that you are doing what they want you to do.
  – … will force you to be more systematic in your development of code.
  – … is more likely to give you a satisfied client!!
One Structured Approach

• A four-step approach to creating a custom application for a client.
  – Requirement Study
  – Prototyping
  – Implementation
  – Structured Testing
Rule Number One

*LISTEN*!!

• Throughout the entire process you need to listen carefully to your client.
• Read between the lines, and ask *lots* of questions!
The Requirement Study: Process

• Conduct a problem analysis by interviewing your clients
• Write a preliminary document
• Present the document to your clients
• Conduct follow-up interviews
• Present a final document
• Be sure that your client is happy with the final document
The Requirement Study: Product

- A written document detailing:
  - Problem Analysis
    - Who the intended users for the system are
    - What the major goals for the system are
  - Product Description
    - What tools will be implemented in the system
    - What all the deliverables are
  - When the system will be delivered
Problem Analysis

• Who are the intended users?
  – What is their level of expertise with software?
  – What is their level of domain expertise?
  – Under what circumstances will they use the system?
  – How often will they use the system?
Problem Analysis

• What are the major goals for the system?
  – Develop a clear statement describing why there is a need for the software system
  – Develop clear sub-goals that can be met with specific software tools
Product Description

• Based on your problem analysis, outline a description of how the system will meet the established goals.
  – For each system sub-goal, create a list of software tools that will meet that sub-goal, and a description of how each tool meets the sub-goal.
  – Create a comprehensive list of deliverables (including software, documentation, support… etc.)
Requirements Study: Presentation

- Submit the requirements study to your clients and give them time to review it (you can sometimes present your study, but try to get the clients a copy beforehand).
- Meet with your client, and discuss the study to ensure that you both understand it completely.
- Once you agree on the final requirement study, set a deliverable date.
NCHS Requirements Study

• Users:
  – Professional epidemiologists and statisticians:
    • mid-level computer expertise, high-level domain expertise, low-level geographic expertise.
    • system use will be semi-regular, in a private, exploratory setting.
NCHS Requirements Study

• Goals:
  – Exploring statistical, temporal, and geographic distributions of mortality statistics and risk factors in order to hypothesize causes of different types of mortality.
    • Sub-goal 1: Compare relationships between a mortality rate and a risk factor, and see how they are distributed in space.
    • Sub-goal 2: View how clusters of high mortality move geographically over time.
NCHS Requirements Study

- **Products:**
  - Software tools
    - Scatter Plot (SP) linked to Map
      - Meets sub-goal 1
    - Parallel Coordinate Plot (PCP) linked to Map
      - Meets sub-goal 1
    - Brushing and focusing in linked views
      - Meets sub-goal 1
    - Simultaneous animation of SP, PCP, and map
      - Meets sub-goal 2
  - Written Documentation
  - Six hours on-site training, up to ten hours phone support.
Prototype

• Develop a rough version of the system:
  – What will the user see?
    • Generate a list of displays that will be needed
  – What will the user do?
    • Generate a list of scripts that will be needed
  – How will the user do these things?
    • Generate a prototype interface, mapping dummy scripts to specific interface tool
  – Implement the prototype in ArcView
NCHS Prototype: Tools

• What will the user see?
  – A map window
  – A Scatter Plot window
  – A Parallel Coordinate Plot window
NCHS Prototype: Scripts & GUI

- What will the user do, and how will she do it?
  - Map a variable – Dialog box called by a button
  - Create a SP – Dialog box called by a button
  - Create a PCP – Dialog box called by a button
  - Brush a window (SP, PCP, or Map) – Tool
  - Focus on the SP or PCP – Tool
  - Play animation – button
  - Step through animation – button
  - Stop animation – button
NCHS Prototype
Implementation

- Write a detailed description of what each script will accomplish
- Figure out what scripts need to be written in what order
- Assign individuals to work on different scripts
Implementation

- Once implementation is completed, you should arrange to demo the product for your clients, and to have them test it for a few days.
- Fix any bugs they find (they will find bugs!)
Structured Testing

- Structured testing is a formal process through which you try to execute every conceivable series of commands. This will ensure that the system does not crash under any foreseeable circumstances.
- This can be accomplished by writing an avenue script that calls different sets of commands, and keeps a log of its activities. If the system crashes, or there are errors, you will be able to determine exactly when they occurred.
NCHS Structured Testing

• Here’s an example script:

'** Testing “Create PCP” and “Brushing”
av.run("Create PCP","Data Table")
av.DelayedRun("Brush",(23@45,40@72),1)

'** Testing a focus event during animation
av.DelayedRun("Play Animation",Nil,1)
av.run("Focus",{64@18})
av.DelayedRun("Stop Animation",Nil,5)

'** Testing the “step” command in the middle of animation
av.DelayedRun("Play Animation",Nil,1)
av.run("Step Throught Animation",Nil)
Conclusion

• The general idea of a structured process are more important than the specific steps in this process.
  – *Think* about your goals and your intended users before you actually start coding.
  – *Communicate* with the intended users throughout the development process.