Choosing Software for Non-profit Organizations

Peter Turk
peter.turk@sumac.com
October 2012

Time Line

Selection Step 1 – Selection Criteria

- Activities:
  - decide who is running the selection process
  - set scope: who will use the software to do what
  - build internal consensus on requirements
  - approximate budget
  - decide whether to buy or build
- Costs:
  - consultants to help select the software
  - staff time

Selection Step 2 – Contact Software Vendors

- Activities:
  - identify those that might do the job for you
  - determine rough pricing and functionality quickly so that you don't waste their and your time
  - *column fodder is nasty*
- Costs:
  - staff time

Selection Step 3 – Demonstrations

- Activities:
  - arrange demonstrations
  - have the right people there
  - allow sufficient time (at least two hours)
  - identify in advance what you want to see
  - *if they won't show it to you, it doesn't exist or is too hard to use*
- Costs:
  - staff time

Selection Step 4 – Check

- Activities:
  - check references (what parts of the software do they actually use, what is good, what is bad)
  - check type and quality of after-sales support
  - check documentation
  - check user groups
  - check organization
- Costs:
  - staff time

Selection Step 5 – Internal Buy-In

- Activities:
  - get agreement from internal interested users
  - get agreement from board
  - inform your IT consultants
- Costs:
  - staff time

Selection Step 6 – Make A Deal

- Activities:
  - negotiate with the vendor
  - sign the contract
  - *vendors usually have a standard contract*
  - *some room for negotiation on services*
  - *price is always negotiable*
- Costs:
  - initial license fee
  - annual renewal fee
  - installation and training fee

Selection Step 7 – Data Conversion

- Activities:
  - get data out of your existing systems
  - put the data into the new system
- Costs:
  - staff time
  - data conversion fees
  - IT consultants

Selection Step 8 – Installation

- Activities:
  - upgrade your computers as required
  - put the software on your computers
- Costs:
  - new computer gear
  - IT consultants
  - database manager license fee

Selection Step 9 – Training

- Activities:
  - learn how to use the system
  - learn how to administer the system
- Costs:
  - travel to training location
  - staff time
  - course fees

Selection Step 10 – Ongoing Operations

- Activities:
  - install updates to the software as they arise
  - retrain staff as new versions are released
  - perform regular data backup
- Costs:
  - IT consultants to (re-)install the software
  - support fees
  - staff retraining time
  - *this is where most of the cost resides*

Software Costs (a DIME)

Development

Introduction

Maintenance

End of Life

Software Costs (1)

- Development
- Introduction
  - network, server, and workstation upgrades: hardware, software, installation, integration
  - purchasing
  - migration (export, convert, import data)
  - installation (server and workstation software)
  - initial training

Software Costs (2)

- Maintenance
  - infrastructure, electricity, cooling
  - security, backup
  - performance: technical (i.e. users waiting)
  - performance: usability (i.e. users figuring out how to do something)
  - upgrades to server and application software (staff and consultant time)
  - ongoing training (staff and consultant time)
- End of Life

Functionality

Functionality – Objective

Manage Relationships With Constituencies

Create Organizational Memory

Benefits

- *efficiency*: Do things more quickly with less effort.
- *effectiveness*: Do things that could not be done before.
- *enhance relationship with community:* Provide better service, reach higher revenues.
- *building organizational memory*: Build an information resource that can be used day-to-day and mined for future work.

Functionality – Basic Efficiency

- Basic data sharing: group contact (client) management
- Automate routine operations:
  - donation processing
  - bulk mailing: mail merge, labels, email

Functionality – Communication

- Relationships between contacts
- Communications: record and manage all types of communications

Functionality – Analysis

- Built-in reports
- Searching for anything you can think of
- *Ad hoc* reporting
- Terminology: knowledge management, decision support

Functionality – Computer Stuff

- Configurability
- Complete import and export (data management)
- Documentation and help
- Security (user profiles)

Functionality – Desirable

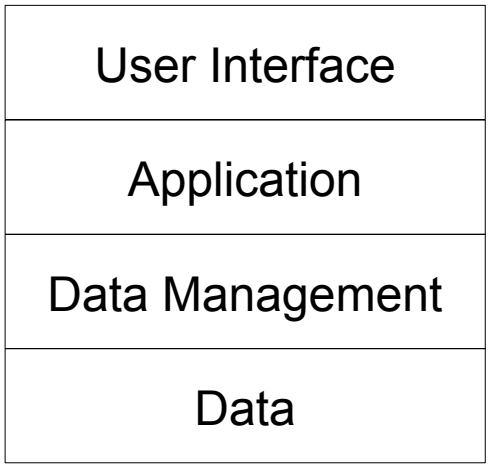
- Campaigns
- Prospect rating
- Reminders
- Moves management (workflow management)
- Mission-specific functionality

Functionality – Integration (No Islands)

- *donate to you*: campaigns, donations, pledges, proposals, prospecting
- *give you money*: auctions, funding programs and requests
- *buy from you*: payment processing, sales, ticketing
- *work for you*: auditions, time tracking, volunteers
- *participate in your activities*: course registrations, events, memberships, submissions, tour booking
- *communicate with you*: communications, contacts, email, reminders

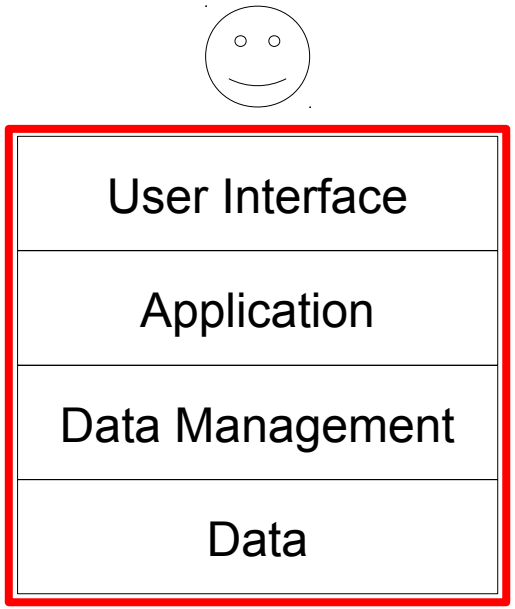
Architecture

Architecture – Terminology



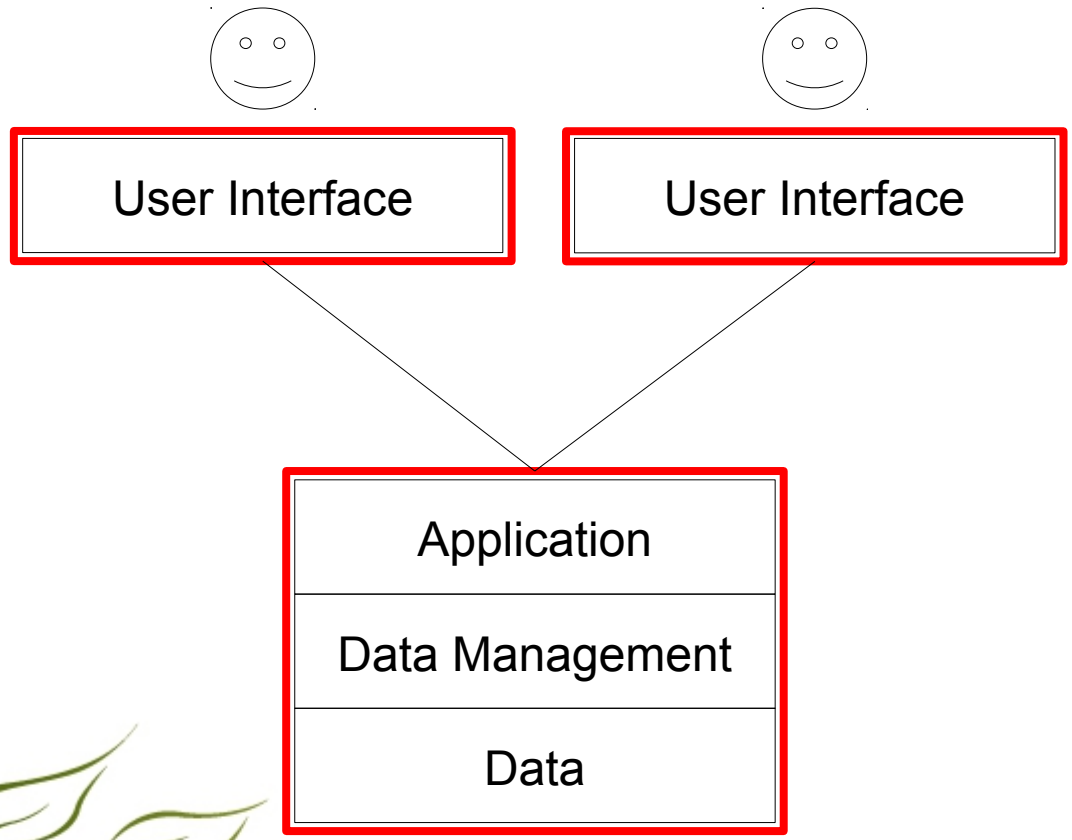
- *user*: you
- *user interface*: presents the application on a display-keyboard-mouse
- *application*: the functionality that you want, e.g. add a contact, delete a reminder, receipt a donation
- *data management*: a database management system (DBMS); usually a relational database using Structured Query Language (SQL) for managing the data, e.g. Access, DB2, MySQL, Oracle, SQL Server
- *data*: the stuff that needs to be stored

Architecture – Single User



Is there more than one person in your organization?

Architecture – Multi User (Hosted)



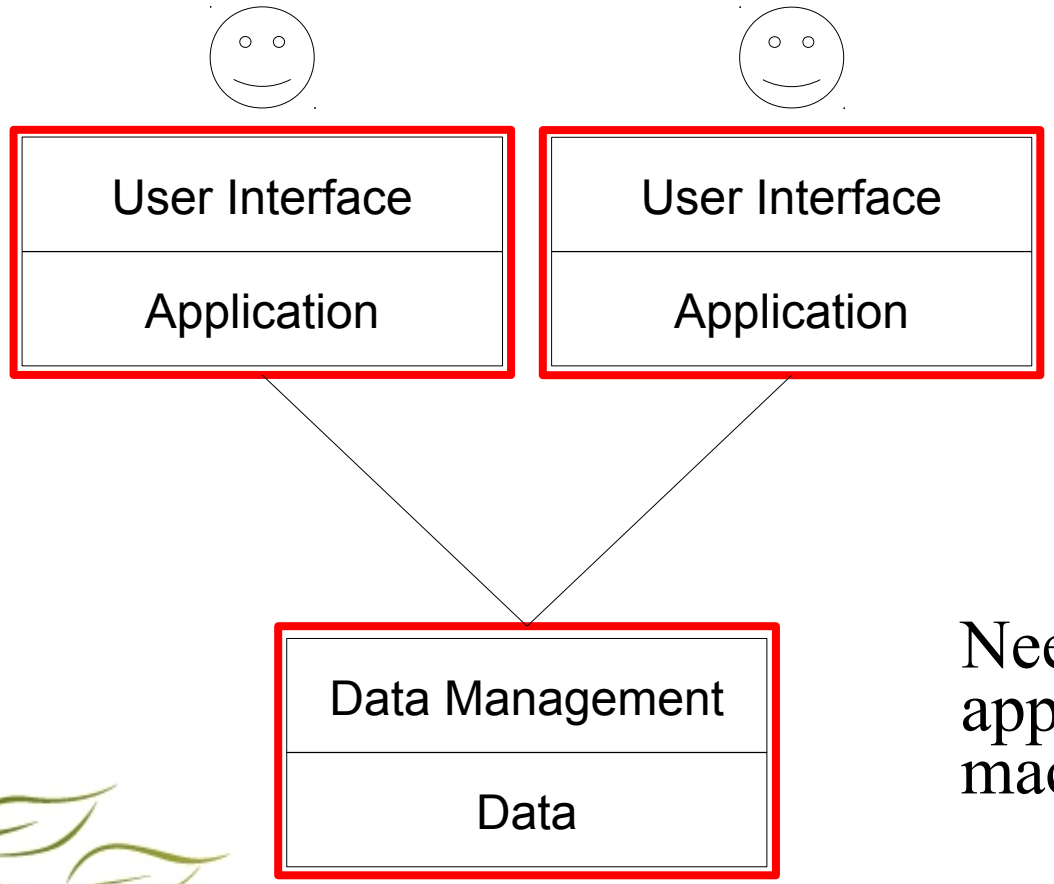
Possibly hosted by an Application Service Provider (ASP) providing Software as a Service (SaaS).

Where is the data? Confidentiality issues, if data is sent over the open Internet or stored outside of Canada. What if your ISP or ASP is down or bankrupt? Can you get your data out?

Costs? Nickles add up.

Usability concerns.

Architecture – Multi User (Client Server)



Need to update the application on multiple machines. How easy is this?

Warnings

Beware

- web sites that purport to be objective but actually front for a software vendor (fundraisingsoftware.org)
- result-selective consultants: (a) pretend to be objective but always recommend the software that they re-sell; (b) cost so much they must recommend expensive software
- hidden fees
- price does *not* indicate what will work best for you
- the answer to "Can it do X?" is always "Yes". Ask:
  - Show me.
  - What does X cost?
  - What would it cost to enable it to do X?

Beware The High Cost of Free/Custom Software (1)

- *documentation*: What will it cost to document a custom package?
- *training*: Who will train you to use a custom package?
- *support*: How fast can you actually get in touch with a university student, during exam time, or Summer when he is planting trees in B.C.? When the student graduates, moves away, or loses interest, all support disappears. A year or two after that the system will stop because of a system software change or need to move to a different operating system.

Beware The High Cost of Free/Custom Software (2)

- *data conversion*: What will it cost to import existing data, clean it up, find duplicates, etc.
- *fees paid to the volunteer*: Often non-zero.
- *inevitable need to switch to another system in two or three years*: This entails real costs. You will need to export the data from your custom system and import it into something else. Is the custom system even able to export data?

Beware The High Cost of Free/Custom Software (3)

- *lack of integration means data will be duplicated*: Volunteers in one database, donors in another, adds data entry burden.
- *time to start*: Custom software must be developed, tested, and re-developed. How much time and effort will it cost you to spend time working with the developer as he gives you new releases, you try them, find them lacking, and cycle back to get another release? This will take many months.
- *data loss*: Almost every in-house developed system we have displaced (there have been lots) has contained bugs that caused it to lose track of data or record things incorrectly.

Success Factors

- Functionality. Access to the functionality (i.e. ease of use).
- Big bang rarely works. Go step by step. Set a small number of basic objectives.
- Organizational desire to use the software and make it work.

Closing