ANOSE Version 1.0 Project Documentation And Real World Results

Presented to the faculty of University of Alaska Fairbanks

In Partial Fulfillment of the Requirements of

MASTER OF SOFTWARE ENGINEERING

By

James Becwar

Fairbanks, Alaska

November 2008

ANOSE Version 1.0
Project Documentation
And Real World Results

By

James Becwar

RECOMMENDED	
Dr. Peter Knoke Advisory Committee Chair	Date
Dr. Orion Lawlor Advisory Committee Member	Date
Dr. Mitchell Roth Advisory Committee Member	Date

APPROVED	
Dept Head, Computer Science Department	Date
Dean, College of Science, Engineering, and Mathematics	Date
Dean of the Graduate School	Date

Acknowledgments:

Melanie Becwar

Thank you for your patience and help with me being in school. Your support has helped so much.

Dr. Peter Knoke

Thank you for your many years of teaching, encouragement and motivation, and for championing the Software Engineering Department at UAF. You have been a phenomenal mentor.

Dr. Orion Lawlor , Dr. Mitchell Roth

Thank you for your help with scoping my project, the encouragement and feedback, and for continuing to share your wisdom with me.

Eric Nace, Jim Rogers

Thank you very much for giving me the resources to be able to complete this project. The business experiences you have shared with me have helped to shape ANOSE quite a bit.

Brad Neufeld

Thank you for help with the database. Your idea of using cross joins to create the order pairs was genius.

Howard Earl

Thank you for the encouragement to work on this project and talking to me about how RDI uses the determinator. It is my hope that project to will allow you to make the determinator even better.

Greg Fischer

Thank you giving the talk about Software Engineering at RDI. The section on statistical evidence gave me the idea to make ANOSE.

Robert & Laurie Becwar

Thank you for many years of support and encouragement. I could not have done any of this without you.

Abstract

A major problem for software development consultants is how to bid a fix price bid accurately. Bids that are too high will not be accepted by the client, and bids that are too low will end up costing the consultant the difference. In this paper we analyze eight factors that are known at the time of bid, in twenty two projects at RDI Fairbanks to see how they affect the actual cost of construction and the stress level of the development team. A data collection, analysis and reporting tool called Analysis of Software Engineering; ANOSE was developed for this project. ANOSE is an open source program available on SourceForge.net for public use.

Several factors had strong effects on the cost of construction, but with limited data, no conclusion could be made for a high confidence level. Adjusting the bid after an initial bid, new development environments and new employees were all strong predictors of high development team stress. A surprising result was that using a waterfall or agile method had no effect on the final cost of the development or the stress level of the team.

Table of Contents

Acknowledgments:....4
Abstract....5
Goals....8
A Real World Example of ANOSE....8
Tools and Methods Used....8
Overview of Factors Analyzed....10
    Waterfall Development Method....10
    Agile Development Methods....10
    Use of COTS Control(s)....10
    A Person with Less than One Year Experience on Development Team....10
    Bid Adjustment for Client after Initial Bid....10
    New Client for RDI....10
    New Development Environment....10
    Remote Client....11
Results....12
  Results for Waterfall Method and Agile Methods....12
  Person with Less than One Year Experience Results....14
  Bid Adjustment Results....16
  New Client Results....18
  New Development Environment Results....20
  Remote Client Results....22
Conclusions....24
 Operation of ANOSE:....25
  Overview of ANOSE Operation:....25
  Related Work:....25
License:....25
Install Dependencies:....26
Install Instructions:....26
  Database Install:....26
  Web App Install:....30
How to Use ANOSE:....34
  User Management:....34
  Setting Up Contact Types:....35
Adding Users to the System:....35
  Setting Up Users:....35
  Setting Up Projects:....36
  Setting Up Questions:....36
    Importance of asking the right questions:....37
    Example Questions:....37
  Answering Questions:....38
  Viewing Results:....38
    One Dimension Plots....38
    Two Dimension Plots....41
  Statistical Analysis:....42
    Sample Summary:....42

One Sample T-Test:....43
Two Sample T-Test:....45
Technical Notes:....46
Database Design:....46
Website Design:....47
Key Classes:....49
Stats Class:....49
SSlTools:....49
Email Class:....49
PolyQuestion:....50
QuestionTypes:....50
DatabaseUtils:....50
Menu Control in detail:....50
Question Controls in detail:....50
PolyQuestion Control in detail:....51
Know issues in ANOSE Version 1.00:....52
Future Plans for ANOSE:....52
References:....53

Goals

The original goal of Analysis of Software Engineering, ANOSE, was to answer a simple question. Does the prescriptive method that James Becwar documented provide statistically significant savings? ANOSE was developed to help organize the data, and automate the statistical inferences. During construction it was realized that ANOSE could be more useful to the software engineering community than the prescriptive method.

The new goal of this project is provide a system that is:

- Able to collect data about any measurable question
- Make it easy to mine the data for information
- Make it easy to test hypotheses
- Easy to use and manage

A Real World Example of ANOSE

This section will present the results of analyzing eight factors that may affect the actual cost of software construction and the stress level of the development team. Twenty two projects that the author worked on while employed with Resource Data Inc., http://www.resdat.com, in Fairbanks Alaska were analyzed. The twenty two projects were split among eight clients. The projects were performed from 2005-2008. All twenty two of the projects were either fixed cost, or time and materials not to exceed projects.

The object of this paper it present which factors affected the budget or development team stress level in a statistically significant way. Some of the factors were known before bidding begin, and it is hoped that this paper will lead to more accurate cost estimation at RDI, by pointing out which factors have the greatest influence on the final cost of construction. The secondary goal of this paper is to give project managers at RDI effective ways to manage development teams stress levels while working on stressful projects. All of these projects had a small budget and were worked on by small teams, normally no more than three RDI employees.

Tools and Methods Used

The budgetary data was collected daily from the RDI timesheet program. All other data was collected in an Excel 2003 spreadsheet. Early in the project it became clear that Excel was not the right tool for this project. In particular there were three major defects in using Excel. The statistical support in Excel was decent, but didn't fit the needs of this project. It was difficult for more than one person to enter data in the Excel spread sheet, and it was difficult to create reports in Excel. A new software package called ANOSE (Analysis Of Software Engineering) was constructed for this project.

ANOSE is a web application (a web site) that allows software engineers to collect information about software projects and do statistical analysis on the data collected. A key goal in ANOSE is to provide a user friendly interface. ANOSE is released under GNU 3 License and is available at http://anose.sourceforge.net/.

ANOSE combines a free database, Microsoft SQL Server 2005 Express, an open source statistical package, R, and a custom web front end to provide a multi-user system that simplifies collecting, analyzing and reporting of the data. The R backend allows for complex statistical queries to be performed, such as a two sample T-Test. The web front end allows for multiple users to enter and query data and reports and includes security features such as users and roles. The database allows for more efficient storage and interoperability than Excel.

ANOSE itself is a fairly small program at roughly 10,000 lines of code with significant synergy between its parts. ANOSE can be queried as data is entered which may allow for more up-to-date data then normally is available to project managers.

Overview of Factors Analyzed

Waterfall Development Method

The waterfall method is an older prescriptive software development methodology. Ten of the twenty two projects analyzed used the waterfall method of software development.

Agile Development Methods

Several agile methods were tested while the author worked at RDI. These include the Spiral method, SCURM, and RAD. All of these methods were combined in to a single category. Twelve of the twenty two projects analyzed used an agile method.

Use of COTS Control(s)

Nearly all software projects use commercial off the shelf (COTS) software such as compilers, code libraries, operating systems, database management systems, and web servers. This factor includes only COTS libraries that provide .Net visual controls, also known as custom controls. Several project managers at RDI avoid the use of custom controls in their projects because of previous bad experiences with buggy custom controls. Eight of the twenty two projects included the use of custom controls. The controls were either commercial, open source, or public domain.

A Person with Less than One Year Experience on Development Team

Having a junior programmer on the development team may cause problems, because of a lack of experience. Six of the twenty two projects had at least one team member with less than one year of experience at RDI.

Bid Adjustment for Client after Initial Bid

Occasionally a client will request an estimate and then after seeing the bid ask us to rework the bid and project scope to make the project fit a smaller budget. The estimation process is often less formal on the second bid. Three of the twenty two projects had their budget and scope adjusted after the initial bid.

New Client for RDI

It has been the author's experience that it is easier to accurately bid on projects with existing clients, because more is known about the environment that the software will be used in and the client's expectations in terms of project deliverables. Ten of the twenty two projects were for new clients. It's important to note that in the case of one client, multiple projects were completed concurrently, serving as the first experience with the client.

New Development Environment

All of the development of the twenty two projects was done using Microsoft Visual Studio 2003, 2005 or 2008. Other projects used a new development environment, such as the Microsoft .Net Compact Framework or the Microsoft Asp.Net Ajax technology. Six of the twenty two projects used a new development environment.

Remote Client

Working with remote clients can be difficult for a variety of reasons. Communications are more difficult, as face to face meetings are rarer which can lead to a multitude of problems, such a failure to understand the problem domain. Four of the 22 projects were for remote clients.

Results

Results for Waterfall Method and Agile Methods

On average both the waterfall method and agile methods came under budget by about 12%. The standard deviation of the waterfall method was 33% and for the agile methods it was 20%. While it appears that the agile methods allow for less variance in bidding, the author of the paper does not believe this is the case at RDI Fairbanks.

Performing a two sample T-Test gave a T-value of -0.00105, with degrees of freedom of 16.6 and a P-Value of 0.999, which lead to the conclusion that it cannot be said that cost savings are not statistically different from one another.

The correlation between using the waterfall method and going over budget is 0.000148 which is effectively no correlation. The agile methods were similar with a correlation of 0.000665.

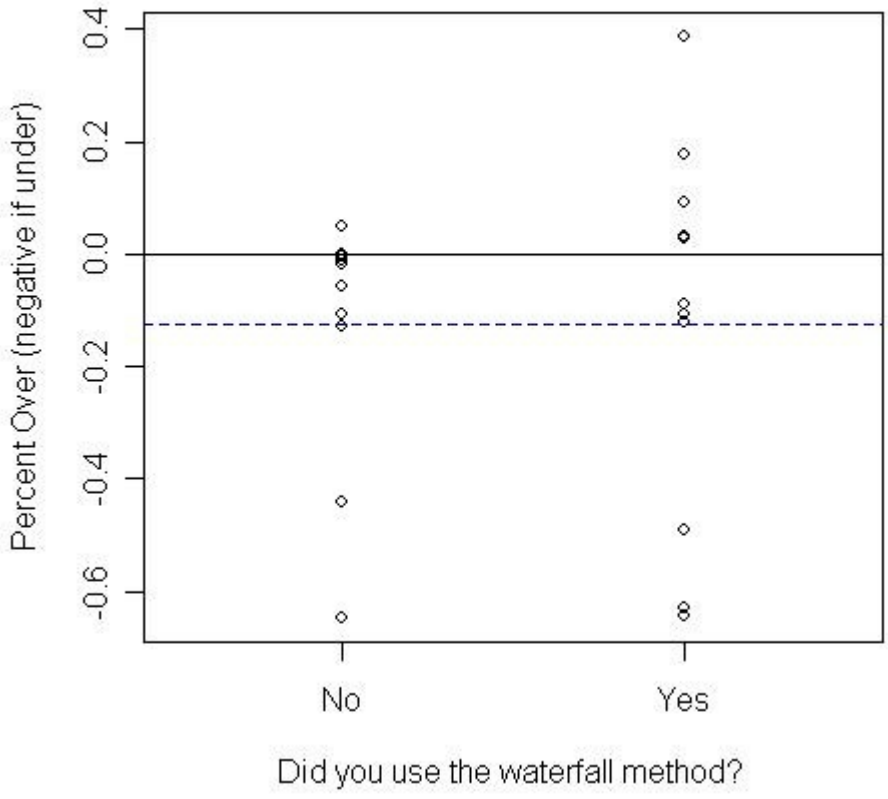


Figure 1– Waterfall vs. Percent Over
Blue dashed line is linear regresion line

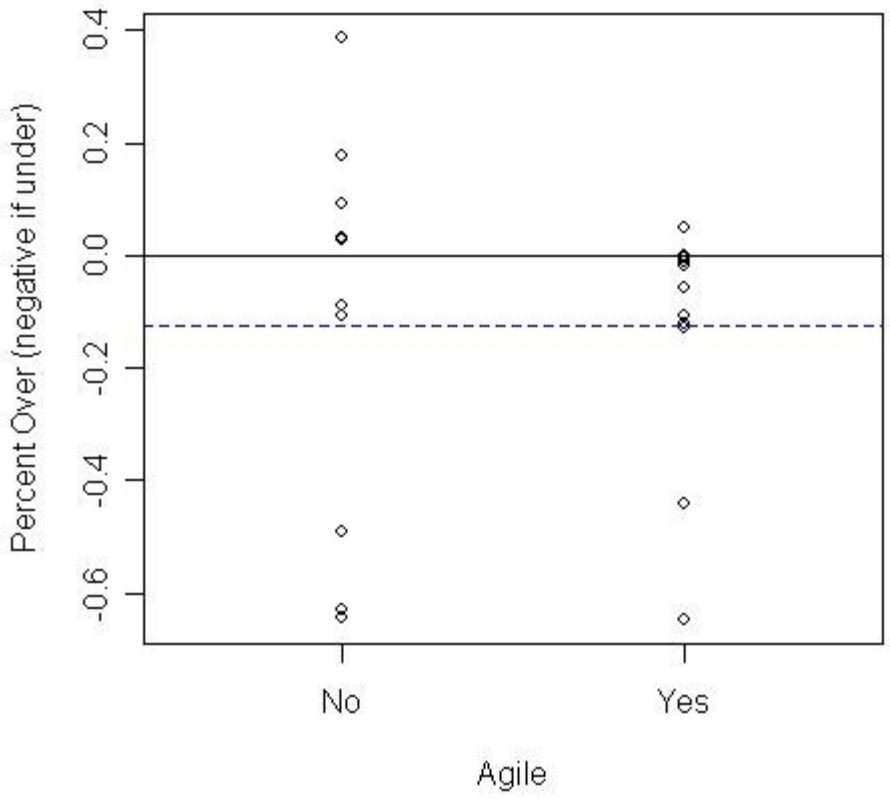


Figure 2– Agile Methods vs. Percent Over
Blue dashed line is linear regresion line

It is important to note that this data is sampled from one company, RDI and cannot be used to make a generalization across the software development industry. Also, there may be sampling error, as RDI tends to use the waterfall method on more complex projects, and agile methods on smaller projects. RDI Fairbanks' workforce is small, currently five developers, therefore the results given may not apply to RDI as a whole or the industry as a whole.

Another aspect analyzed is the effect of development methodology and developer stress. The waterfall method had a correlation of 0. There was very small negative (-0.069) correlation between using the agile method and the stress level. While the correlations are very small they are worth noting, as team stress can be difficult to manage.

It is the recommendation of the author that for small projects, project managers at RDI use whatever development methodology they and their team feel comfortable with, as at this point there is no strong evidence that the development mythology has a strong effect on going over budget.

Person with Less than One Year Experience Results

New employees can bring much needed labor and fresh ideas, but also make it hard to budget the cost of construction as each programmer produces work at different rates depending on their skill level and experience. It also takes time for new employees to become accustomed to the corporate environment and the resources that are available to them.

RDI has a unique solution to this problem by using a concept of non-bill time. RDI allows all employees to non-bill time to a client. When the client receives their bill, the hours are noted but not charged for. New employees are expected to have some non-bill while they get up to speed on the project and technology being used.

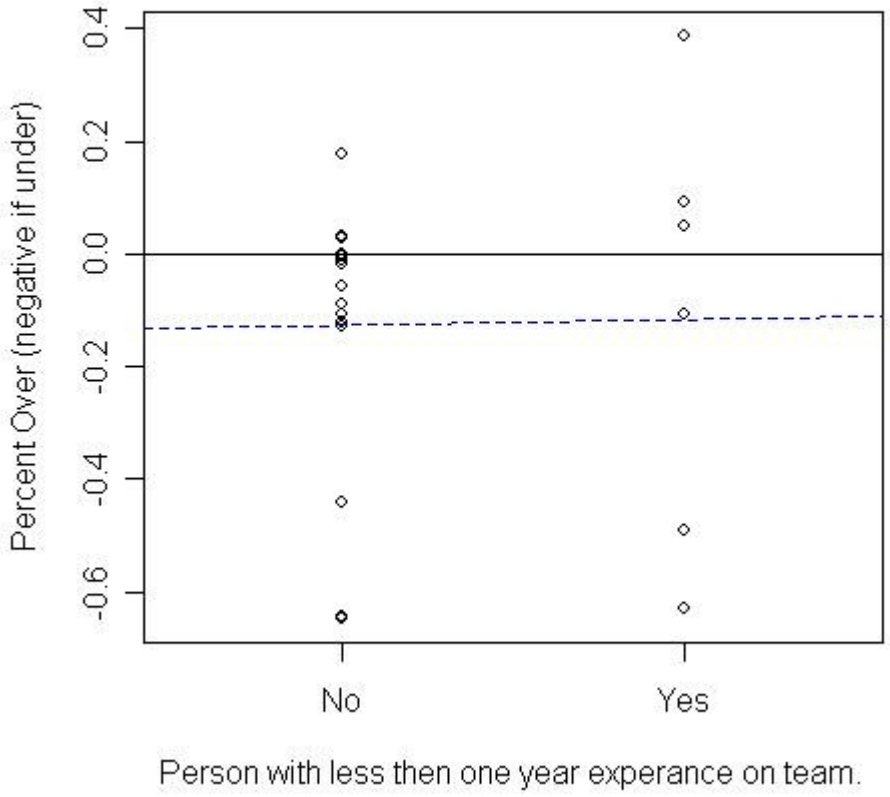


Figure 3– New Developer on Team vs. Percent Over
Blue dashed line is linear regresion line

On average, projects without new developers are 12% under budget with a standard deviation of 23%. Projects with new developers were on average 11.6% under budget with a higher standard deviation of 38%.

The correlation between having a team member with less than one year of experience and going over budget was calculated to be 0.0180 which is a very small correlation.

Performing a two sample T-Test resulted in a T-value of 0.0651, degrees of freedom: 6.53 and a P-value of 0.950. From this data it cannot be said that the means are statistically different.

It is the conclusion of the author that at RDI Fairbanks, the non-bill policy appears to be fulfilling the company's commitment to it's clients that new employees will not affect the budget of projects bid on.

While comparing new developers and team stress level a moderate correlation of 0.386 was noted. If the project manager was concerned about the development teams stress level, the author of this paper would recommend not adding developers new to RDI.

Bid Adjustment Results

On three of the twenty two projects, the budgeted amount and project scope was adjusted after an initial bid to the client.

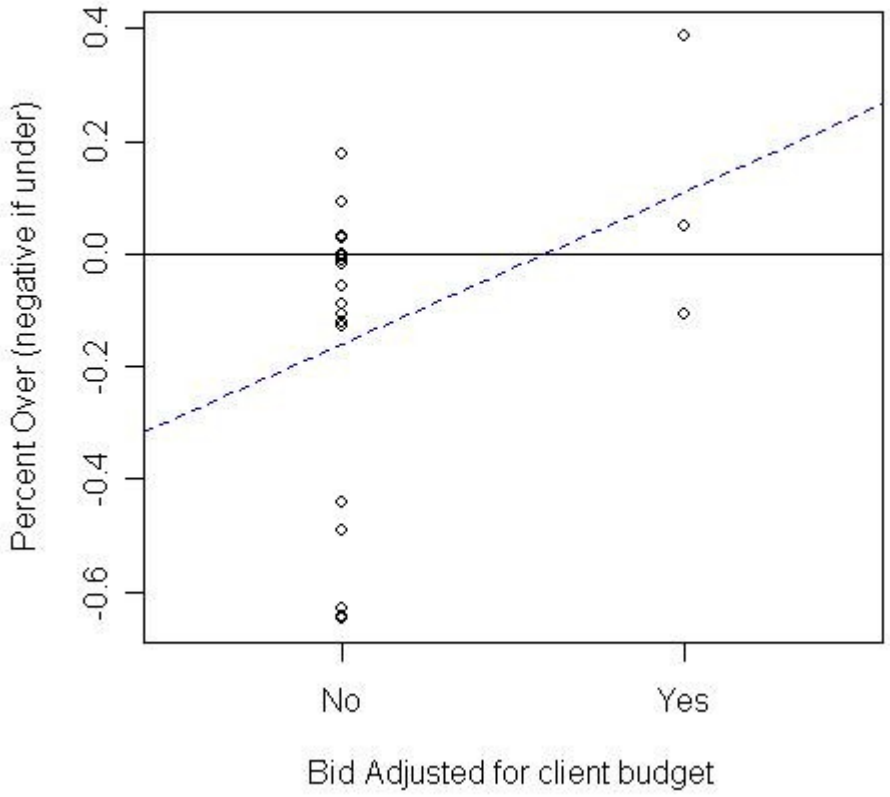


Figure 4 - Bid Adjustment vs. Percent Over
Blue dashed line is linear regresion line

Projects that did not have an adjusted bid on average were 16% under budget with a standard deviation of 26.5% percent. Projects that did have an adjusted budget were over budget by 11% on average and had a standard deviation of 25%.
A moderate correlation of .346 was noted between adjusting the budget and going over budget.

Performing a two sample T-Test produced a T-value of 1.71, Degrees Of Freedom of 2.75 and a P-Value of 0.193. At an 80% confidence level it can be said that the two means are statistically different.

It's important to note that only three samples had adjusted bids, so future data may change this result. The author of this paper would caution project managers at RDI Fairbanks to be aware that there has been a history of overages when adjusting the budget

after an initial bid. There is insignificant data to say that practice should be avoided all together.

There is strong correlation of .525 between bid adjustments and high development team stress. A T-value of -7.10 with degrees of freedom of 18 and a P-value of 0.00000129 was calculated when performing a two sample T-Test. With such a small P-value it can be said at a 99% confidence level that the average stress level is different when the budget is adjusted.

Projects managers at RDI Fairbanks should be cautioned that there is a strong correlation between development team stress and adjusting the bid after the first bid. The author of the paper would like to have more data before suggesting any kind of policy change.

New Client Results

Ten of the twenty two projects were for new clients. Note that for one client, because no projects had been completed for the client previously, three projects starting simultaneously were considered the first projects.

Projects with existing clients were on average were 9% under budget with standard deviation of 25%. Projects with new clients on average were 16% under budget with a standard deviation of 30%.

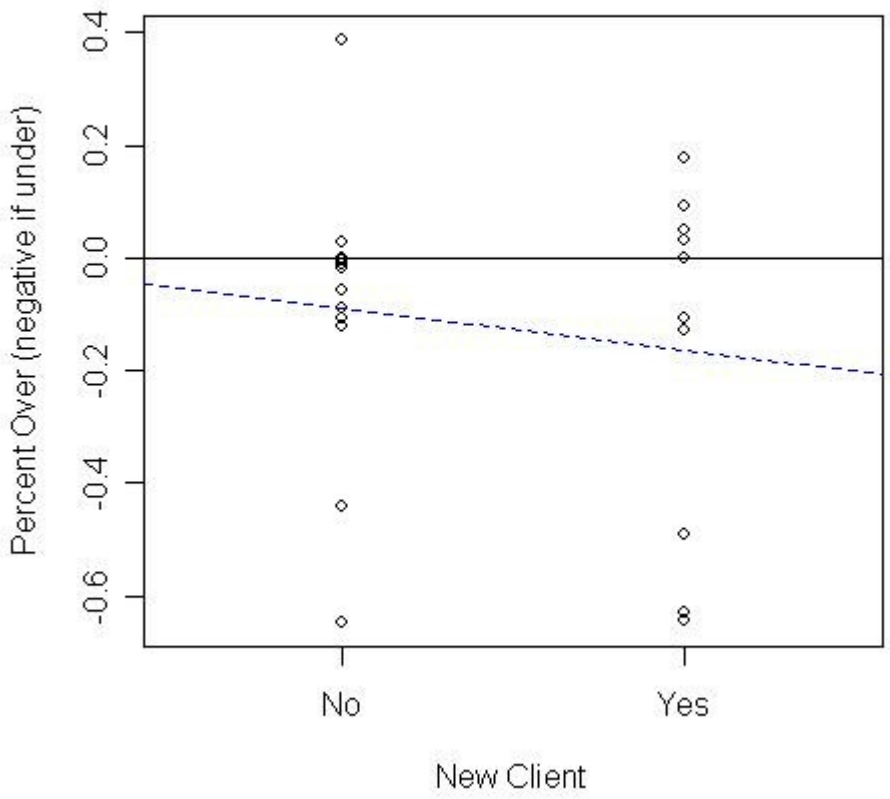


Figure 5 – New Client vs. Percent Over
Blue dashed line is linear regresion line

A weak negative correlation of -0.138 was noted between new clients and going over budget.

Performing a two sample T-Test resulted in a T-value of -0.613 with Degrees of Freedom of 17.5 and a P-Value of 0.547. From the data collected it cannot be said that the average overage is different for new clients and existing clients for reasonable confidence levels.

The week negative correlation suggest that there may be some over bidding with new clients, but two sample T-Test does not show a statistically significant difference. Collecting more data may reveal if project managers have a habit of over bidding new clients.

There is a week correlation of .259 between new clients and high development team stress. A T-value of -1.18 with degrees of freedom 17.9 and a P-value of 0.253 were calculated from a two sample T-Test.

It cannot be said that the average stress level is different between new and existing clients at a very high confidence level, but there is some indication that projects with new clients are more stressful then projects with existing clients. RDI project managers could use this factor to help manage stress levels on the development teams.

New Development Environment Results

On six of twenty two projects, the development team used a new development environment, such a new version of Microsoft Visual Studio, a new version of the .Net framework, or a major change in development patterns, such as using Microsoft Asp.Net AJAX technology, or the .Net Compact Framework on handheld computers.

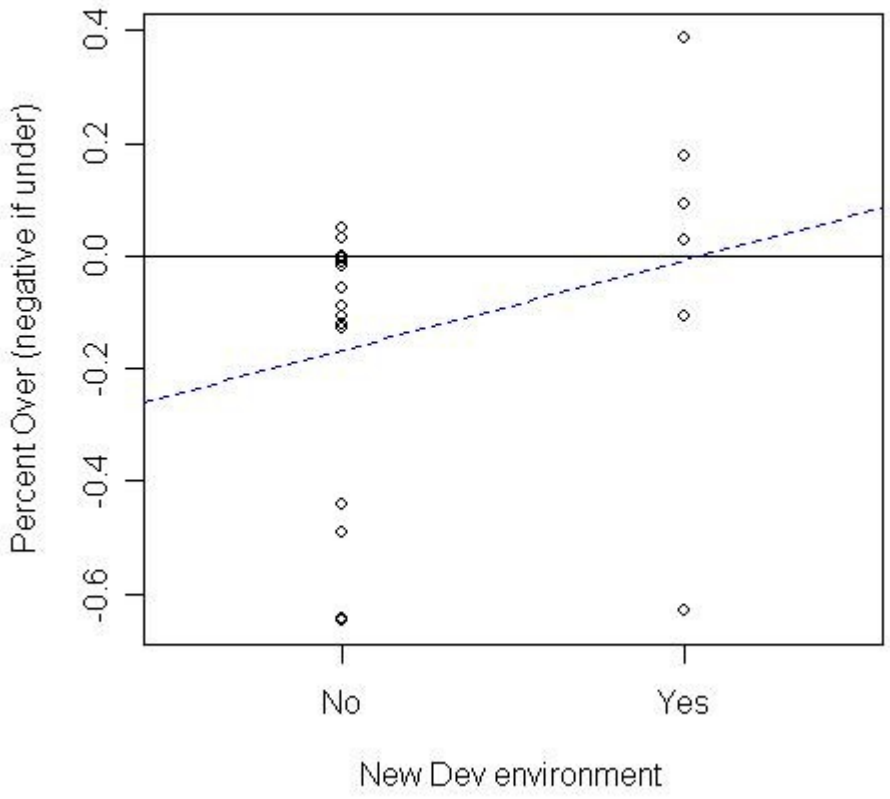


Figure 5 – New Development Environment vs. Percent Over
Blue dashed line is linear regresion line

When a new development environment is used the project is under budget by 0.9% with a standard deviation of 35%. This compares to the average of 16% under budget with a standard deviation of 24%.

Performing a two sample T-Test gave us a T-Value of 1.03 with Degrees Of Freedom of 6.93, and a P-Value of 0.336. It cannot be said that the means are statistically different at relatively low confidence levels.

A small correlation of .263 was noted between using a new development and going over budget.

A moderate correlation of 0.386 was noted between using a new development environment and high developer stress.

The results of a T-Test produced a T-value of -1.75 with degrees of freedom of 8.00 and a P-value of 0.119. It can be said at an 80% confidence level the average developer stress level is different when a new development environment is used.

Remote Client Results

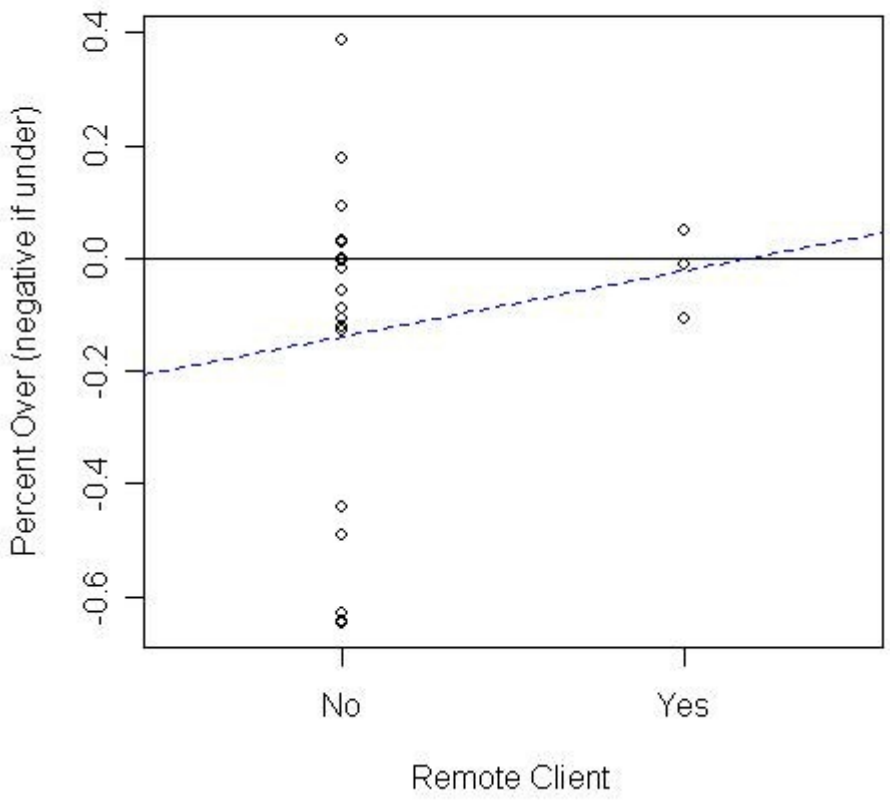


Figure 6 – Remote Client vs. Percent Over
Blue dashed line is linear regresion line

Three of the twenty two projects were for remote clients. Projects with local clients were on average were 14% under budget with a standard deviation of 29%. Projects for remote clients were on average were 2% under budget with a standard deviation of 7%.

Performing a two sample T-Test produced a T-value of -1.45 with degrees of freedom of 13.3 and a P-value of 0.171. It can be said at an 80% confidence level that the means are statistically different.

A small correlation of 0.149 was noted between being a remote client and going over budget.

While there is some indication that projects with remote clients tend to be closer to going over budget than projects with local clients, the author would like to caution the

reader that there were only three projects for remote clients, which could have led to an artificially low standard deviation.

When comparing remote clients and developer stress level a small correlation of 0.250 was noted. Performing a two sample T-Test resulted in a T-value of -1 with degrees of freedom of 2.45 and a P-value of 0.405. With the data currently collected it cannot be said that there is a statistical significant difference in developer stress level between remote clients and local clients.

Conclusions

While no silver bullet was found for cost estimation during the bidding process, several factors, known at the time of the bid, seem to influence whether a project will be over or under budget. However, in most cases, more data should be collected before policies can be created.

Adjusting the bid a second time seemed to have a strong negative effect as did using new development environments, but in both cases more data is needed before a firm conclusion can be reached. Project managers should be aware that there is evidence, but no firm rules can be stated.

There is some evidence that projects for new clients are being over bid. More research would be need to determine if this is true, and by how much. Project managers should be notified of this fact, as this may give RDI a competitive edge with new clients, but project managers should also be cautioned to not be overly optimistic with new clients.

Strong evidence was found that remote clients are being under bid, but only three projects were for remote clients, so no firm conclusion can be reached. Again, project managers should be notified of the result, but more data will be needed before a policy can be made.

Strong evidence was found also that development methodology and employees with less than one year of experience did not significantly affect the chance of going over budget. If a client is concerned that a new employee is going to increase the cost, we can use these results to show that the non-bill policies work and that the project's budget will not be affected. Project managers should also be encouraged to use whatever development methodologies they and their team feel most comfortable with.

The relationships between stress and the factors analyzed were more pronounced. Adjusting the bid a second time, new development environments, and having new employees on the development team were all strong predictors of development team stress. Project managers should be cautioned that these factors contribute to development team stress, and the project managers should take necessary steps to reduce these factors.

The development methodologies used, a remote client, or a new client did not significantly affect the stress level of the development team. This information could be encouraging to project manages, as many project managers are hesitant of remote projects, but this information suggests RDI would be well served by seeking out remote clients as a way to increase the potential market that RDI has available to it.

Operation of ANOSE:

Overview of ANOSE Operation:

Once ANOSE is setup and configured, an administrator will create users. Users, in this system, are called contacts and are organized by contact types. Some examples of contact types are programmers and project managers. The administrator would create questions relating to the project and then specify which contact types they should be directed to. This feature is beneficial because some questions may not be within the scope of all contact types and this would save time and reduce cost of running ANOSE. The administrator would also add projects to the system, so that contacts would be prompted to answer questions relating to one or more projects.

Any question that has a numerical answer or questions that can enumerated, such as a yes/no, can be asked in ANOSE. Some questions that might be entered into ANOSE are:

- What was the project budget?
- Was this a high stress project?
- Did you use the waterfall method?

After the questions are answered, the administrator would login to the system and begin seeing results.

The ANOSE system uses the statistical engine R to provide visualization of the data collected and the relationships between multiple questions, and to perform statistical analysis based on the answers.

Related Work:

At this time, the author of this paper has been unable to find any works similar to ANOSE. Several sources can be found, which describe using statistics in software engineering [1][2][3][4]. However, no programs that help collect, display and analyze the data for the software engineering field could be found. It is possible that many researchers use general purpose software such as Excel and S+ to collect and interpret their data. The method of data collection seems to vary with the project. The idea of a web based software product like ANOSE has been around for at least ten years, but the software tools available on the web were too primitive to implement [5].

License:

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

Install Dependencies:

The following must be installed on the ANOSE web server before ANOSE's install can begin:

- Windows 2000, XP Pro, Vista Business with IIS server installed and configured
- .Net 2.0 and ASP.Net Configured for IIS
- R version 2.6.0
- R (D)COM wrapper 2.5S
- Microsoft SQL Server 2005 Express configured for SQL Server authentication and TCP/IP connections.

Install Instructions:

Database Install:

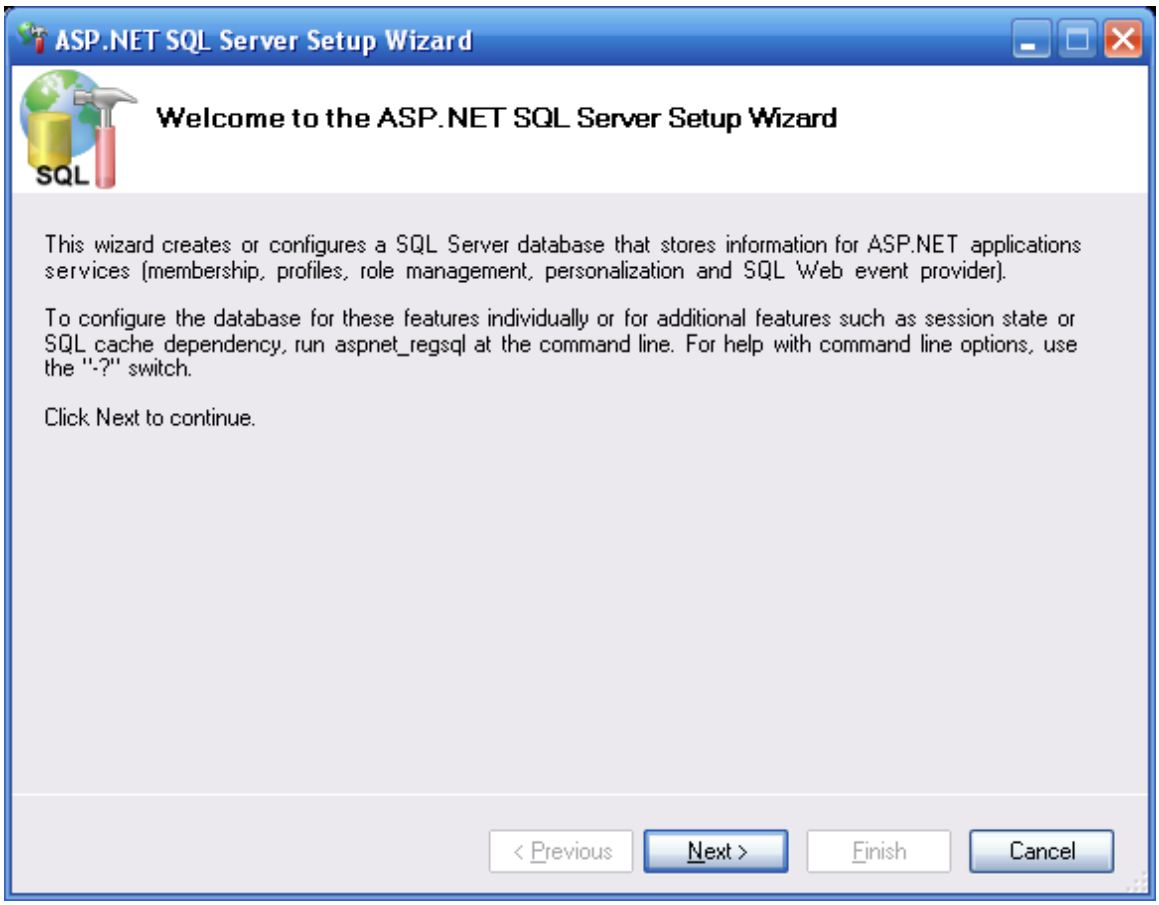
Once the dependencies are installed on the server, the administrator must create a local SQLServer user that ANOSE will use on the database. The user should have select/update/insert/delete rights on all of tables in the ANOSE database. The user name and password for this account will be required for future use. The administrator should then create a new database for ANOSE to use.

Next, the aspnet_regsql.exe program must be run to setup the database to work with the ASP.Net Membership utilities. From a command window the directory must be changed to:

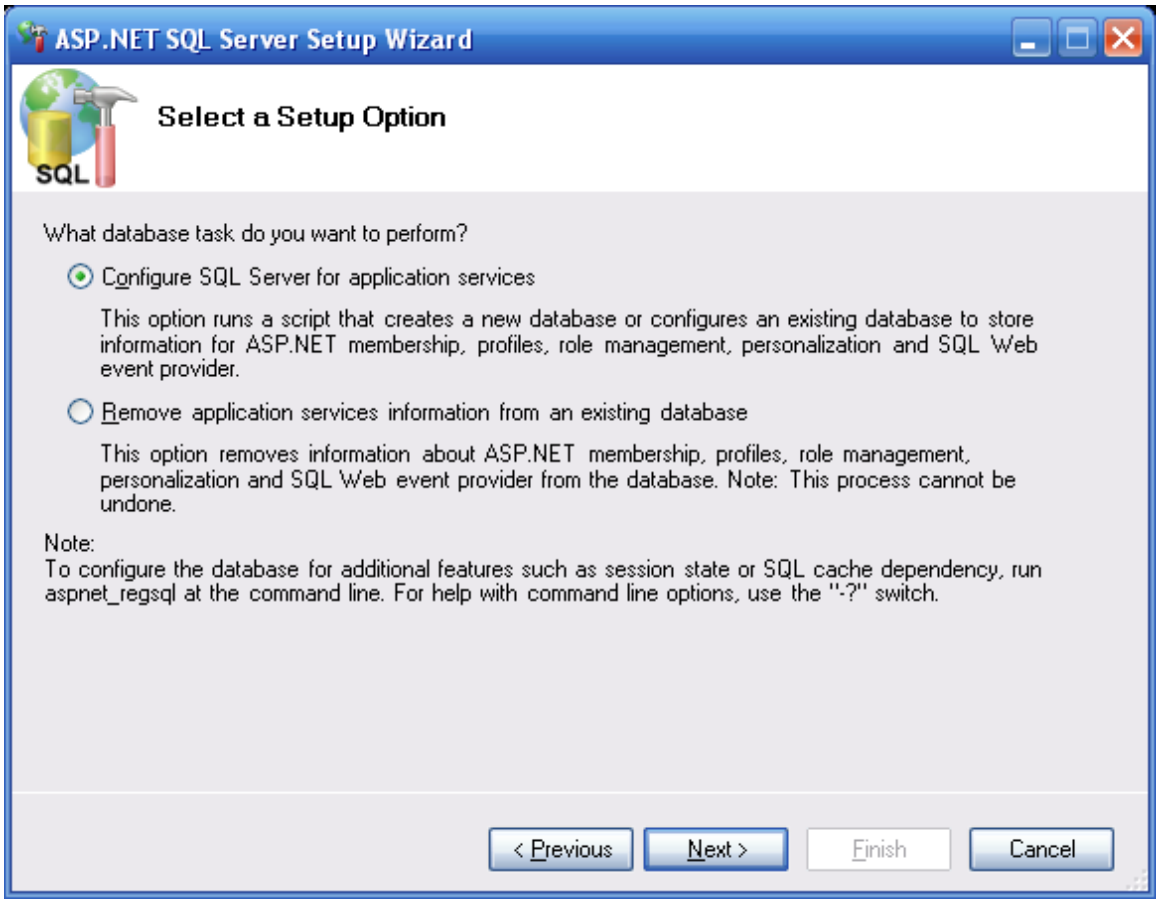
\windows\Microsoft.NET\Framework\V2.0.50727

And run the aspnet_regsql.exe file (Included .Net version 2.0)

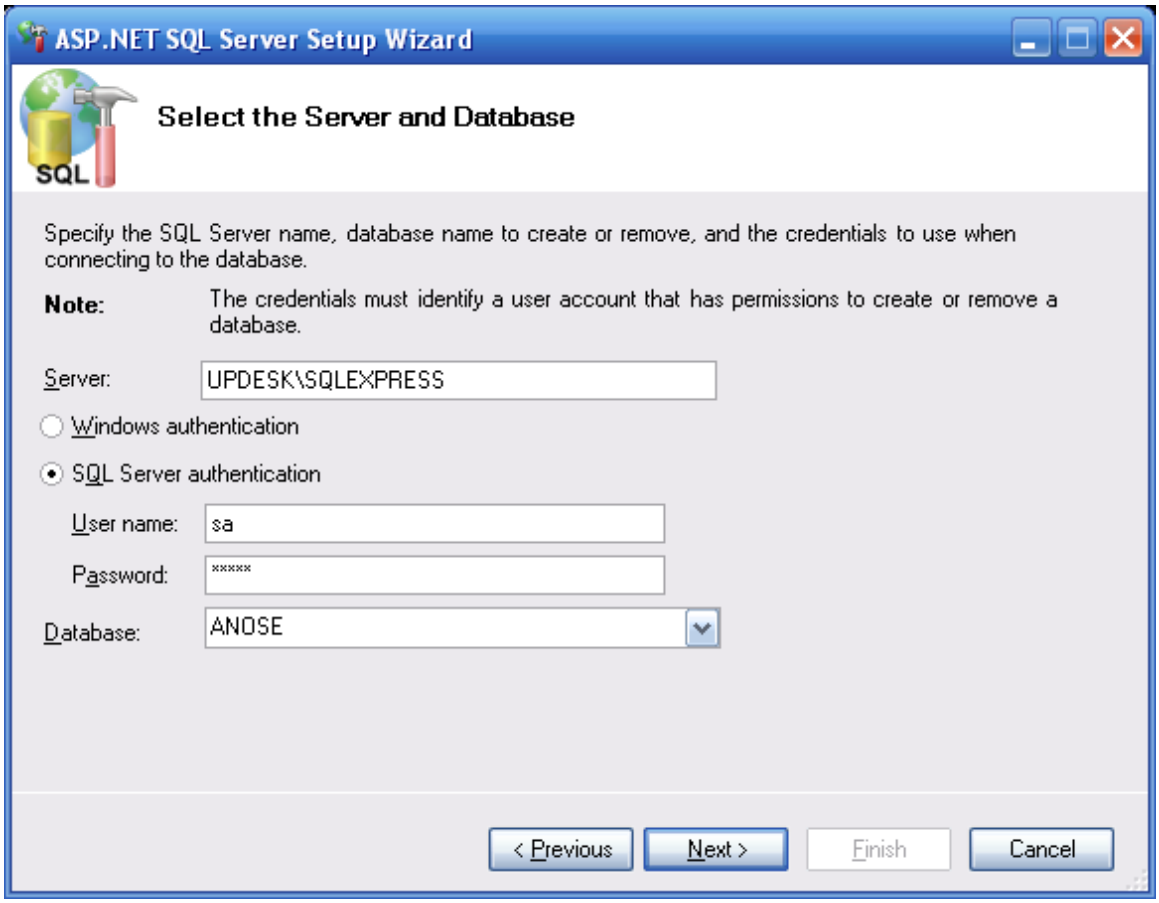
If completed correctly, the image shown below should appear:


ASP.NET SQL Server Setup Wizard
Welcome to the ASP.NET SQL Server Setup Wizard
SQL
This wizard creates or configures a SQL Server database that stores information for ASP.NET applications services (membership, profiles, role management, personalization and SQL Web event provider).
To configure the database for these features individually or for additional features such as session state or SQL cache dependency, run aspnet_regsql at the command line. For help with command line options, use the "-?" switch.
Click Next to continue.
< Previous
Next >
Finish
Cancel

After clicking Next, the administrator should select the top option “Configure SQL Server for application services” in the window shown below.



The administrator should enter the server name, select SQL Server Authentication, enter the database user name and password and select the ANOSE database.



After clicking Next, the administrator should review the summary and click Next again.

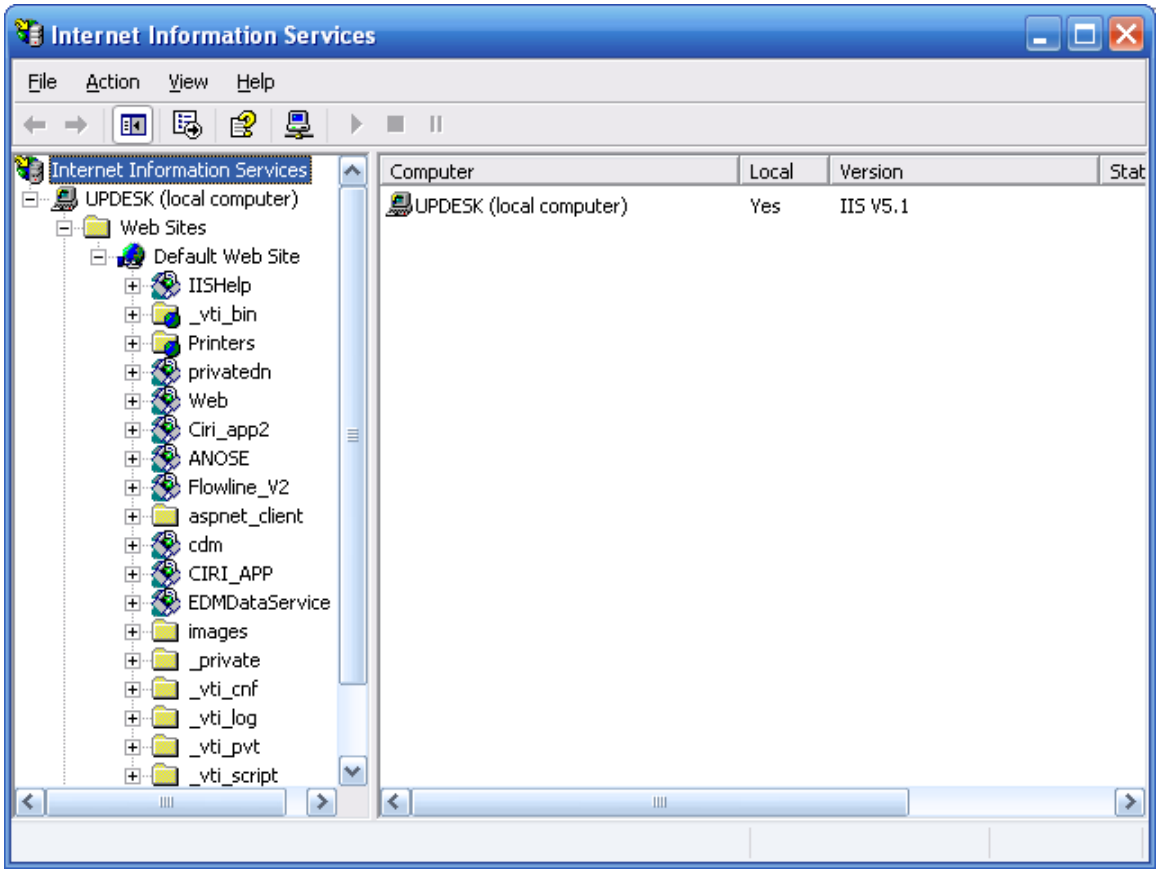
The window shown below should appear and the administrator should click Finish.

After the membership tables are created, the administrator should run the DBInstall.SQL script to create the necessary database objects.

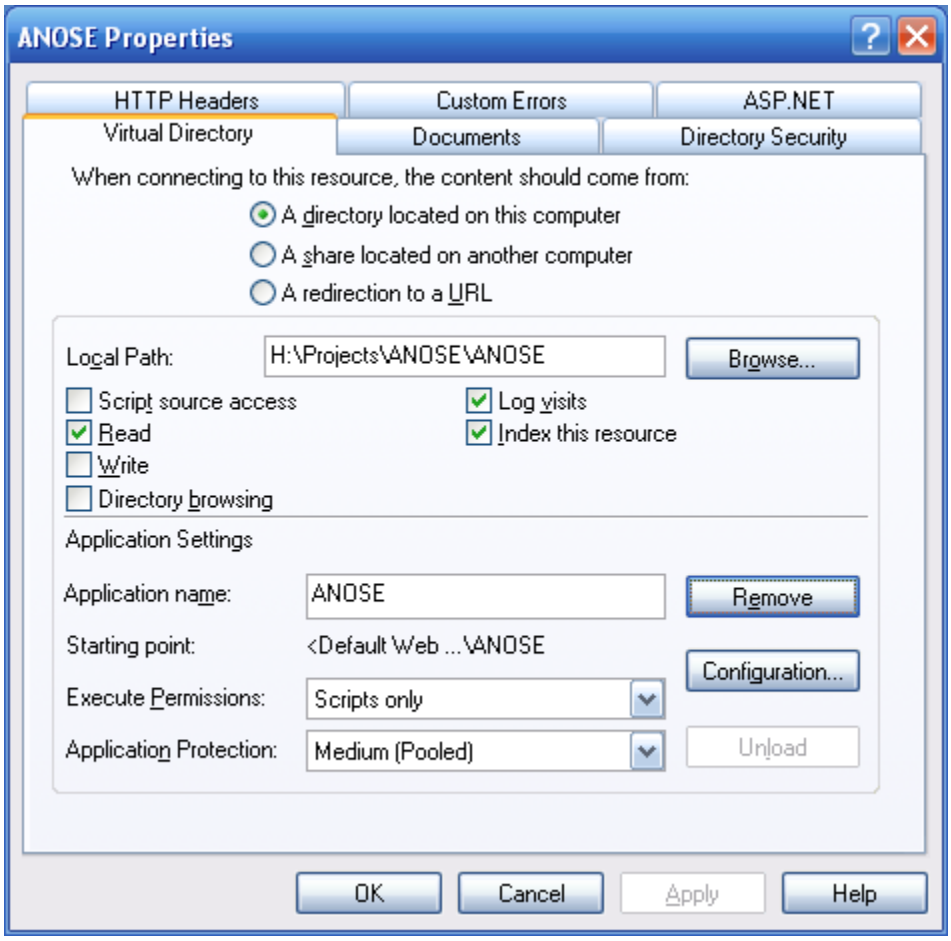
Web App Install:

The web application is located in the ANOSE folder of the install disk. The folder ANOSE_Website should be copied to \Inetpub\wwwroot.

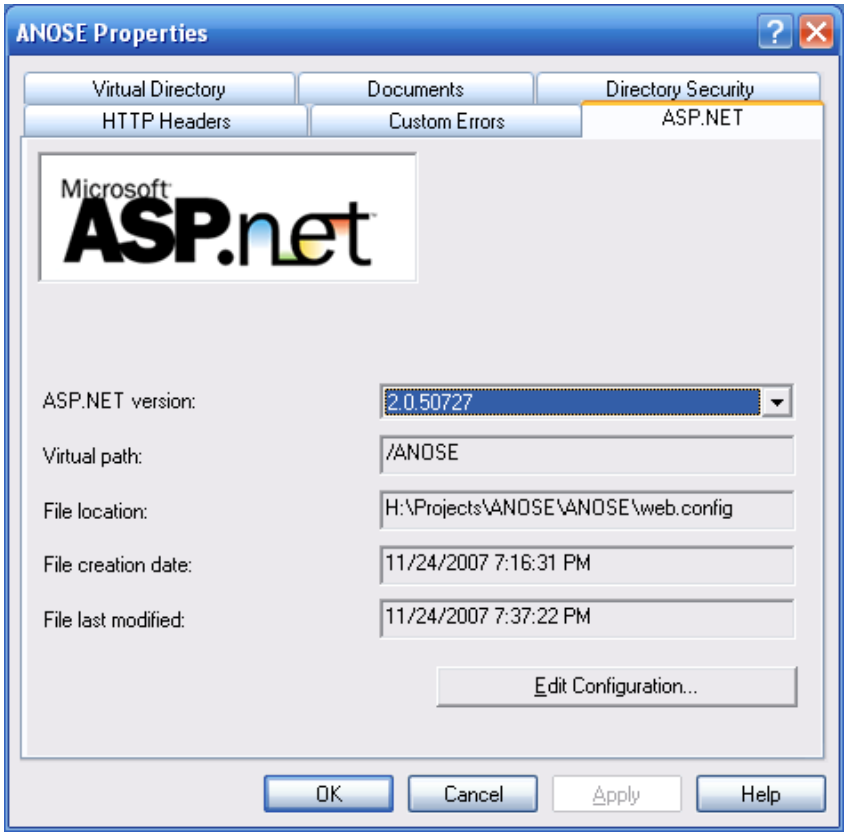
Next the Internet Information Services MMC plug-in, located in the control panel, should be started. The local computer, websites, and default web site tree items on the left hand side should be expanded so that it resembles the image below:



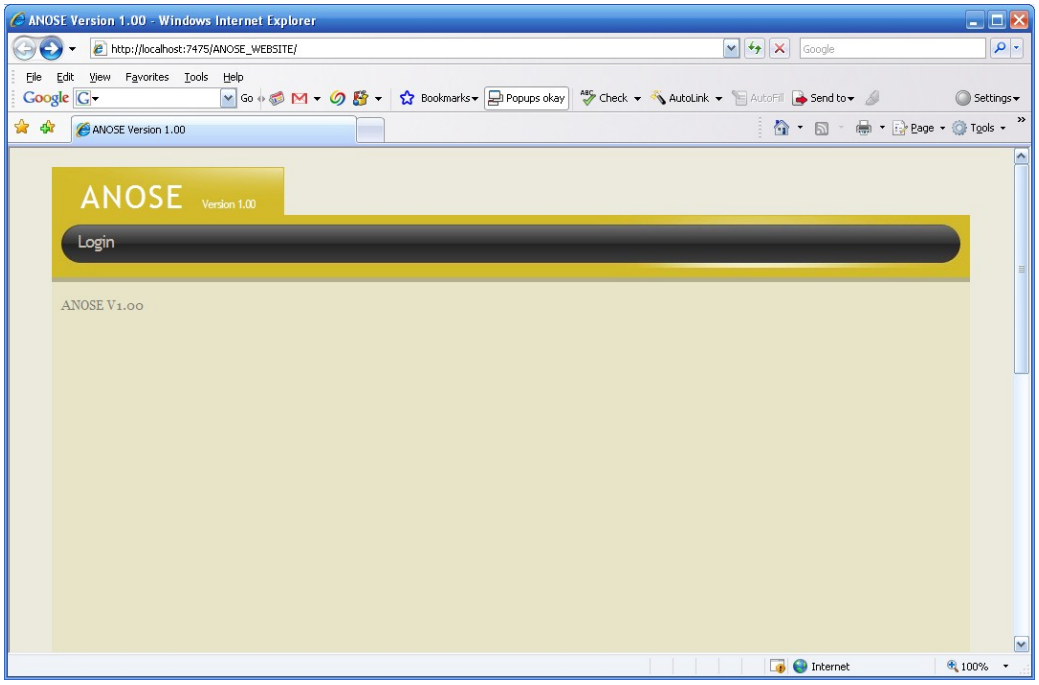
The administrator should then right click on the ANOSE item and select Properties from the popup menu. On the Virtual Directory tab, Remove should be selected under the application name, if available. After this, the administrator should click create. The page should look like:



On the ASP.NET tab, the ASP.Net version should be 2.0.50727 as below:



After clicking OK, ANOSE should be available on the IIS server, and the ANOSE website should appear in the internet browser as shown below.



Before a user can login to ANOSE, the web.config file must be edited. The web config file is located at \inetpub\wwwroot\anose\web.config.

The text MY_DB_USERNAME should be replaced with the database user name and likewise the text MY_DB_PASSWORD with the database password.

There are no default logins in the ANOSE system. To bypass the security system, the web.config file must be edited as shown below.

Replace near the bottom:

```
<location path="Admin">
        <system.web>
            <authorization>
                <allow roles="Admin"/>
                <deny users="*"/>
                <allow users="?"/>
            </authorization>
        </system.web>
    </location>
```

With:

```
<location path="Admin">
        <system.web>
            <authorization>
                <allow users="*"/>
            </authorization>
        </system.web>
    </location>
```

In order to create a user, the administrator should go to http://localhost/ANOSE/Admin/Membership.aspx, and follow the instructions to create the role: admin. Next, a user should be created and then assigned to the admin role.

The web.config file will need to be restored to re-enable security, in order to prevent unnecessary access to admin functions.

How to Use ANOSE:

User Management:

The user management screen is located under the admin menu and has the title of Manage Users. The box under User List is all the logins in the system. Roles and Users can be edited or deleted from this screen. Additionally, roles can be added or removed from users by first selecting a user, and then scrolling down to the Roles section and clicking the box representing the desired action.

To create a new role, the administrator would type the role name in to the box next to the Create New Role button and then click Create New Role.

To add a new user to the system, the administrator would scroll down to the bottom of the screen to the New User section and fill out the appropriate boxes and then press the Create New User button. After the user is created, the administrator would need to assign them any desired roles.

Setting Up Contact Types:

Contact types determine where questions are directed. For example, project managers may get asked different questions than the programmers, so that users are not asked questions outside of their scope of work on any given project.

To add or remove contact types, the administrator would login as an Admin, go to the admin menu, and click Manage Contact Types. Under a section titled New Contact Types for Projects, there should be a text box to enter the new contact type name.

After typing the new name of the contact type, clicking the insert link button should allow the administrator to view the user type under the Existing Contact Types data grid. To change an existing user type, click edit. Clicking delete will delete the user type from the ANOSE system.
Note: You can not remove a contact type that has contacts assigned to it.

Adding Users to the System:

Users are users who can login to ANOSE. Users can be members of security roles, such as the Admin role. Note that users are not the same as contacts. Users are a security construct, a login, and Contacts are users who answer questions in the system.

Setting Up Users:

Contacts are users who are involved in a project and who questions will be directed to.

To add, edit or remove Contacts, the administrator must first login as an admin, click the admin menu and then click Manage Contacts.

To add a Contact, the administrator would enter in the employee's first name, last name and email address, and select their UserID from the drop down list. If their UserID is not listed it's because they have already been assigned to a contact and are in the list of Existing Users. Once the required information has been entered, clicking on the insert link will add them to the system.

To edit an existing contact's information, the administrator could go to the Manage Contacts screen, click on the edit link in the row of the user being edited and click update after making any necessary changes. A contact can be removed by clicking on delete next to the contact name. However, in order to completely remove a user/contact from the ANOSE system, the login account must also be deleted.

Setting Up Projects:

ANOSE can collect and interpret data from multiple projects. In order to view information about projects individually, Project profiles must be created. Questions which are entered into ANOSE will be customized for each project. For example a question might say, for project X what was the budget? In each individual instance, X would be substituted by the project name.

To manage projects, the administrator would login as an admin, click the admin link and then click on manage projects.

To add a new project, the name of the project should be typed under the section that says Add New Project and then insert should be selected. After creating a new project, it should appear in grid view under Projects. To edit or remove an existing project, choose edit or delete respectively. Note that projects that have recorded answers to questions may not be deleted.

The administrator would also be able to manage which contact types questions are directed to on a project to project basis. The administrator would select the project from the project list and scroll down to the Manage Questions section. There is a row for each Question and User Type combination. The administrator would click edit on a row and then check or uncheck the ask checkbox. If the ask checkbox is checked, the question will be asked for that contact type.

Another aspect of project management is determining what each contact's type is in a particular project. An employee may be a project manager and a programmer on one project, but not a project manager on another. The administrator can change a contact's type by selecting the project from the projects list and scroll down to the Users In Selected Project section. The contact may be edited or deleted from this project in this section as well. The administrator could also add a contact to a project by selecting the contact and contact type(s) from the Add User To Project section on that page. It is important to note that a contact may have more then one role in a project.

Setting Up Questions:

Any question that has a numeric answer or an answer from a countable set, such as Yes/No, can be used in ANOSE.

To setup a question, the administrator would login as an Admin, then go to the admin menu and click on manage questions. This will open a section that will display all previously created questions. The administrator would scroll to a section titled Insert New Question and under text, would enter the question desired. From the Question Type drop down, the administrator would select which type of question it will be. The currently available question types are:

- YesNo – Only allows Yes/No Answers. Stored in the database as 1 for Yes, 0 for No
- Scale – Allows number between min and max
- Selection – Allows users to select from a selection

- Number – Allows users to enter any decimal number
- Money – Allows users to enter a money value
- Date – Allows users to enter a date

Next, the administrator would enter a min value and a max value. These values are only used for scale questions. After clicking insert, the question will be added to the ANOSE system and made available for use. Note that new questions will not automatically be assigned to existing projects, and must be manually set up for use in these cases.

Importance of asking the right questions:

Please note that it is **VERY** important that the questions entered into ANOSE are selected carefully. Asking the "wrong" questions can at best create worthless data, and at worst, contribute to reaching incorrect conclusions.

Also, you must realize that if you are using the built in statistical functions to ANOSE, there is no way of comparing more than two question's answers at a time. In other words, there are no transitive relationships between answers.

For example, say the administrator wished to determine whether using an Agile method contributed to going over or under budget. The administrator might have chosen the following questions:

- What was starting budget?
- What was the final cost?
- Did you use an Agile method?

Using the questions above, ANOSE cannot answer the question, because no answer directly showed whether the project budget was over or under. To answer the above question the first two questions would need to be combined. This would change the list to:

- What was the final percent over or under budget? Enter negative for under budget and positive for over budget.
- Did you use an Agile Method?

With this data, ANOSE could do a T-Test to see if there was a statistically significant difference.

Example Questions:

Here are some example questions that you could ask using ANOSE:

- What was the final percent over or under budget? Enter negative for under budget and positive for over budget.
- Did you use an Agile Method?
- Did you use a prescriptive Method?
- Did you use any formalized method?

- Did you write the test plan before construction?
- On a scale of one to ten how complete were the requirements documents? 10 being completely documented and 1 being no documentation.
- On a scale of one to ten how secure do you feel the system is? 10 being totally secure, 1 being very unsecure.
- On a scale of one to ten how stressful was working on this project? 10 being totally stressful, 1 being no stress.
- From the following list, please select the project manager on this project.
- From the following list, please select the client for this project.
- Did you use object orientated programming on this project?
- Did you use a function programming language like C on this project?
- Did you use a new product on this project?
- Did you use a COTS product in development on this project?
- On a scale of one to ten, with one being low quality and 10 being the highest quality, how would you rate the quality of the software developed on this project?
- Did you use a relational database management system on this project?

Answering Questions:

To answer a question, a user would log into ANOSE, select Answer Questions from the menu. If the user has no pending questions, the page will appear blank. Pending questions will appear in a list with project names listed next to the questions. The user would click on the select link and the question would be displayed below. The user would enter the answer and press the answer question button. Due to a bug in .net 2.0, the answer question button must be pressed twice. Future versions may not have this problem.

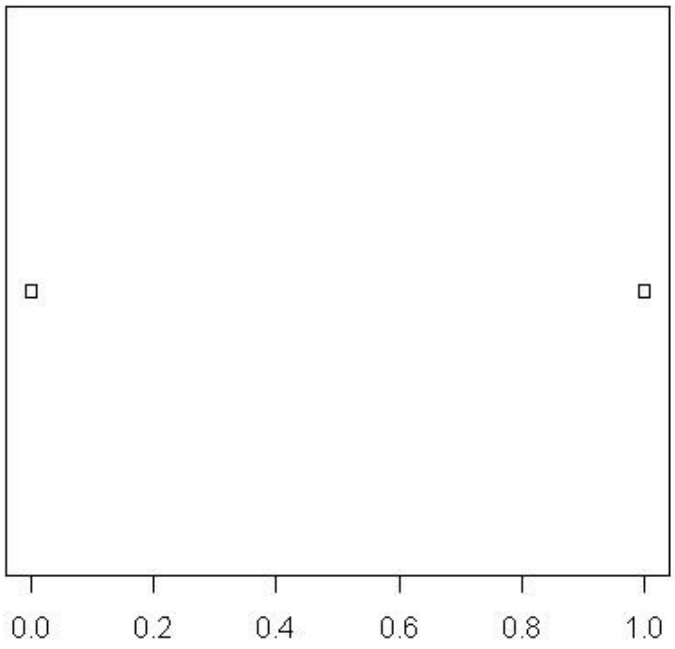
Viewing Results:

ANOSE uses R to generate graphics from the data collected. While R has a rich graphics library, only a subset is currently exposed in ANOSE. The graphics are subsequently not publication quality, and are designed to be used only to support analysis of the data and inferences.

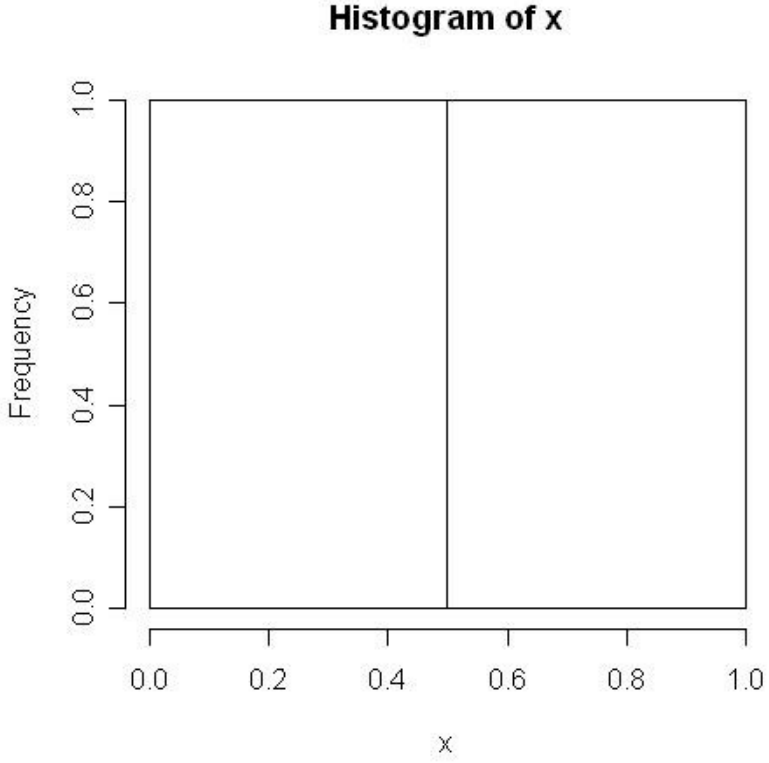
One Dimension Plots

One dimension plots in ANOSE are plots of answers to only one question. The plots include histograms, box plots, and strip charts. Here is an example of each one using the same simple dataset:

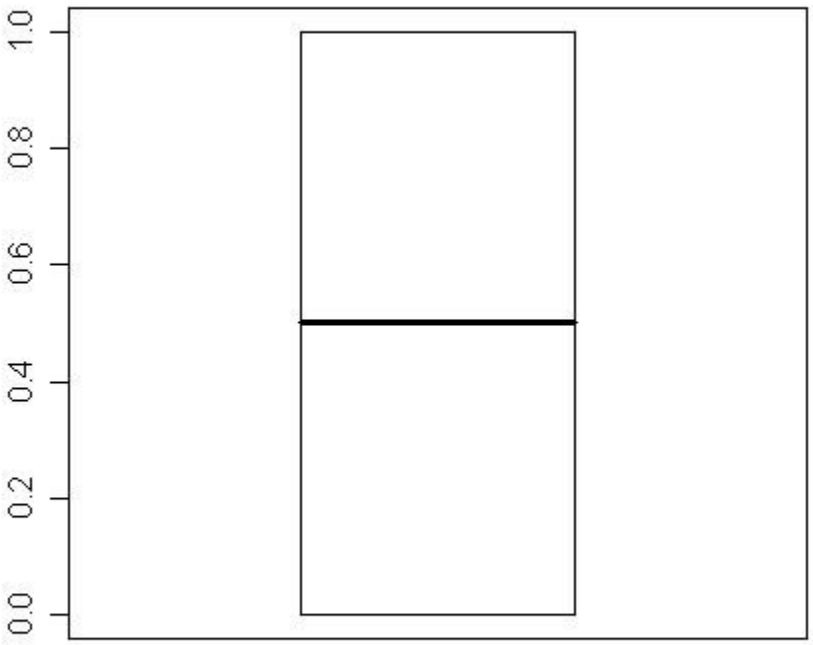
This is a strip chart of the example question “Did you use an agile method?”. You can see there is one yes answer (1.0) and one no answer (0.0). Strip charts are used to get a special view of the data and to look for clumping of data.



Next is a histogram chart of the same question. You can see we have one answer in the [0,.5] and one answer in the [.5,1] range. Histograms are useful to look for a skew in the data.



Finally we have the histogram. From the histogram below we can see that the average answer is .5 by the location of the solid black line. We can also see that the standard deviation is slightly less then the interval 0 and 1. Histograms give a good view of averages and can help detect a skew in the data and help identify outliers in the data collected.



Two Dimension Plots

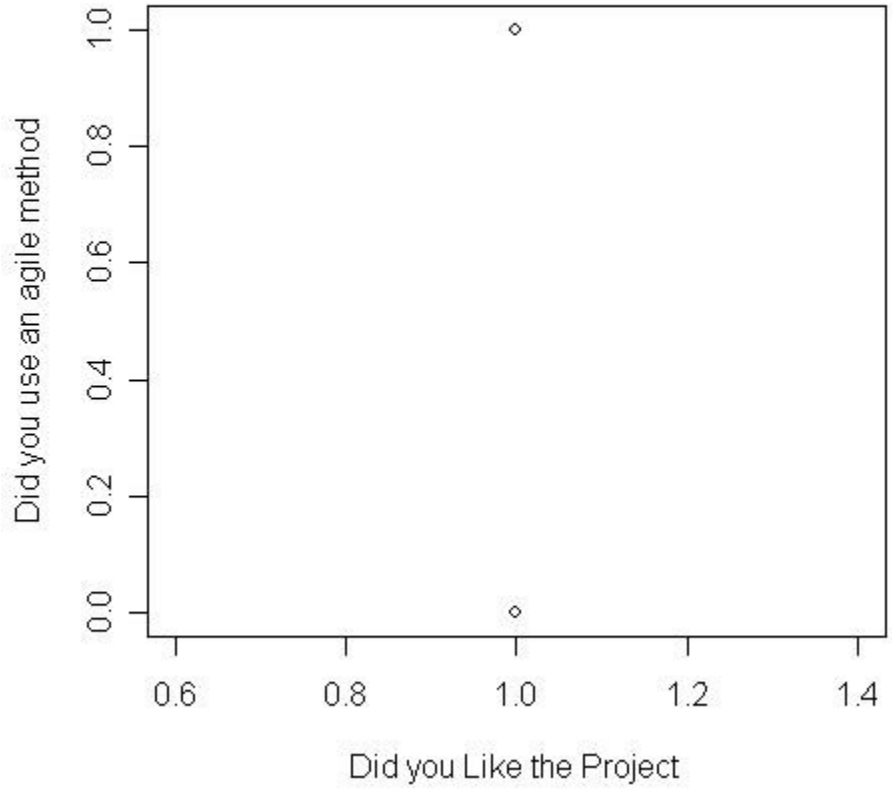
ANOSE allows users to create two dimensional plots by selecting two questions and relating the answers based on the same project. The actual SQL query used to generate the data is:

```
SELECT Q1.Answer as X, Q2.Answer as Y
        FROM ANSWER Q1 CROSS JOIN Answer Q2
        WHERE Q1.PROJECTID = Q2.PROJECTID and Q1.QuestionID = @QOne and
Q2.QuestionID = @QTwo
```

And if a single project is selected, the following is appended:

```
and Q1.ProjectID = @ProjectID
```

Below is and example graphic. In the example data there are two projects. On one project, an agile method was used, and on the other it wasn't. In this case the user liked both as the result was 1.0 on each.



Statistical Analysis:

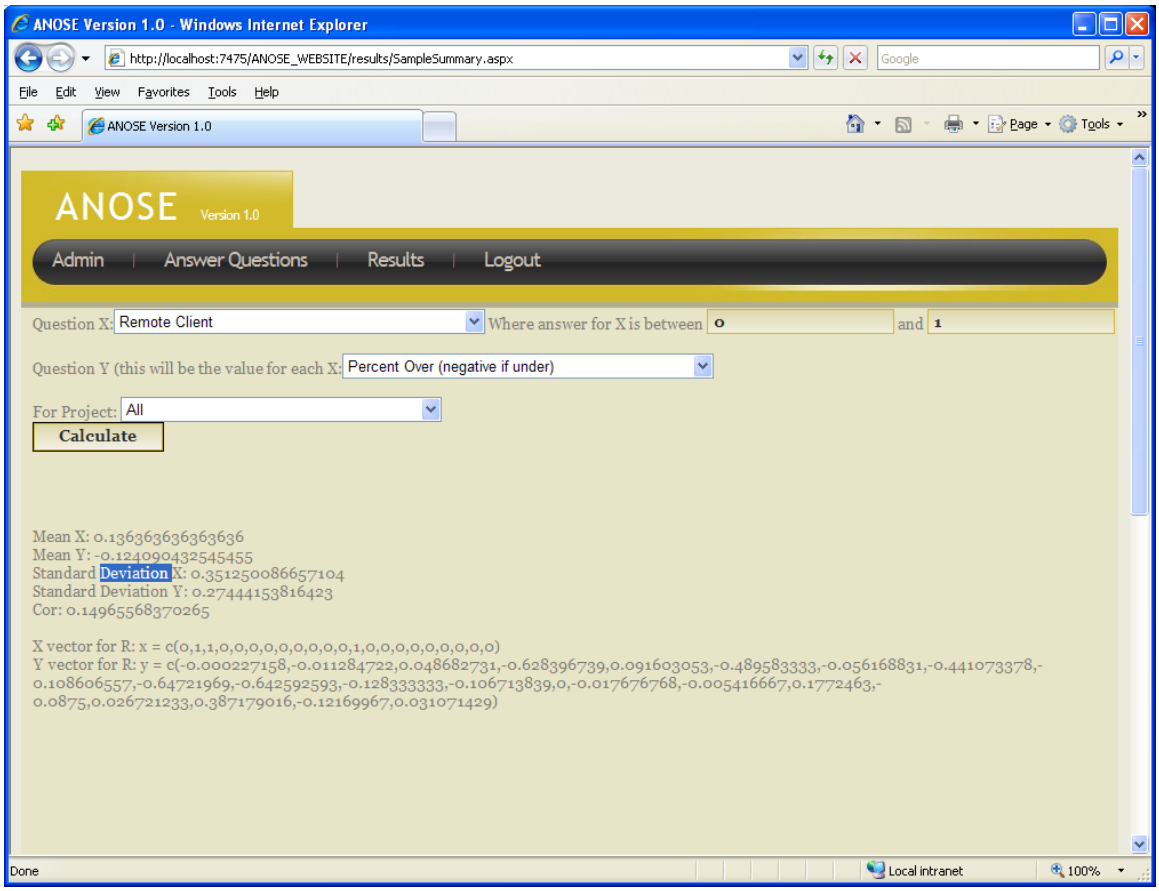
A note on statistical analysis: It could be very easy to use the powerful tools that ANOSE provides to reach incorrect conclusions. Users would be well advised to make sure the underlying conditions for each test are met before using the results that ANOSE provides. Also, users may wish to have a statistician review conclusions.

Sample Summary:

The sample summary allows the administrators to display the following for two related vectors X and Y:

- Mean and Standard Deviation of X values
- Mean and Standard Deviation of Y values
- The correlation between X and Y
- The actual X and Y vectors calculated

To do this, the administrator would login as an admin, go the results menu and select Sample Summary. The screen display should resemble the one shown below.



Question X is the sample from the population (an example question might be "Did you use an agile method."). An administrator could limit the values of X if a subset of the answers to X, such as only the "Yes" answers to the former example question, was desired.

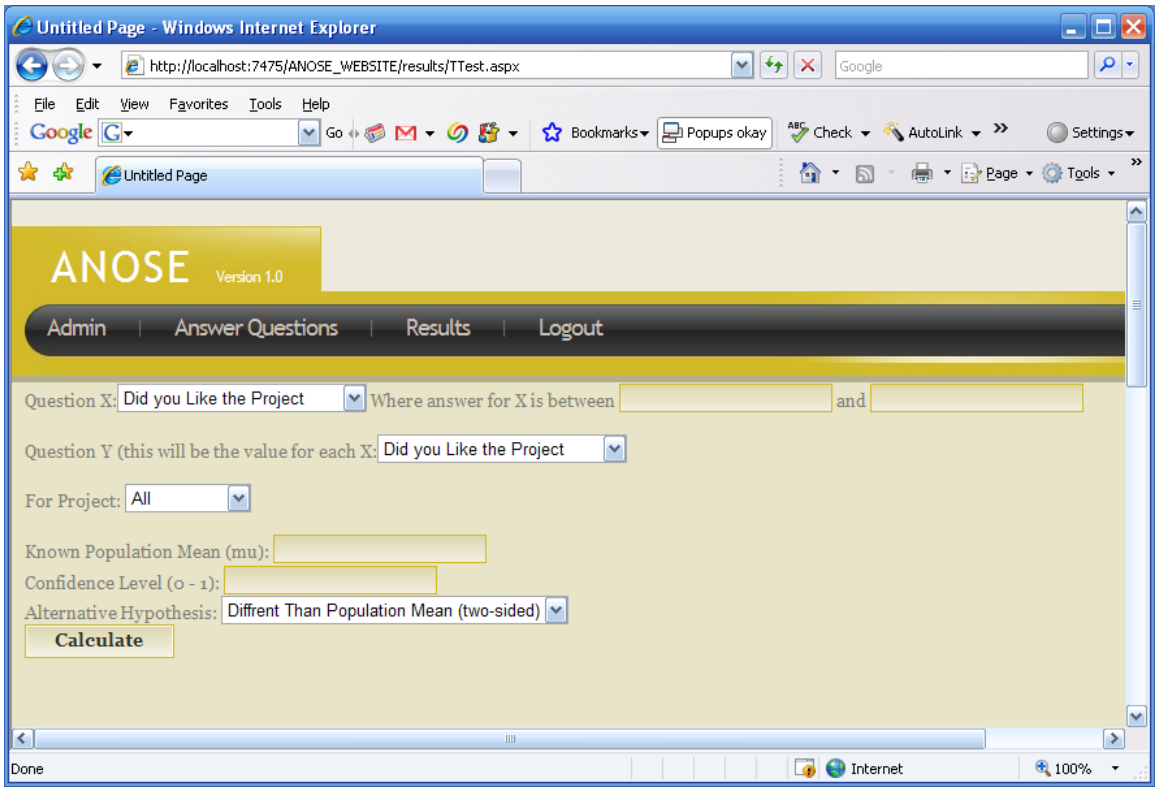
Question Y is the value of each X. Normally this would be related to budget, time or stress.

The "For Project" dropdown is another limit on the X values, you can analyze one project or all projects.

One Sample T-Test:

One Sample T-Tests are used to compare a sample to a known population mean (mu), such a published average. Normally, the population mean is unknown, so the Two Sample T-Test is more regularly used. The Two Sample T-Test is covered in the next section.

ANOSE allows users to easily perform a One Sample T-Tests. To do this, the administrator would login as an admin, go the results menu and select One Sample T-Tests. The screen display should resemble the one shown below.



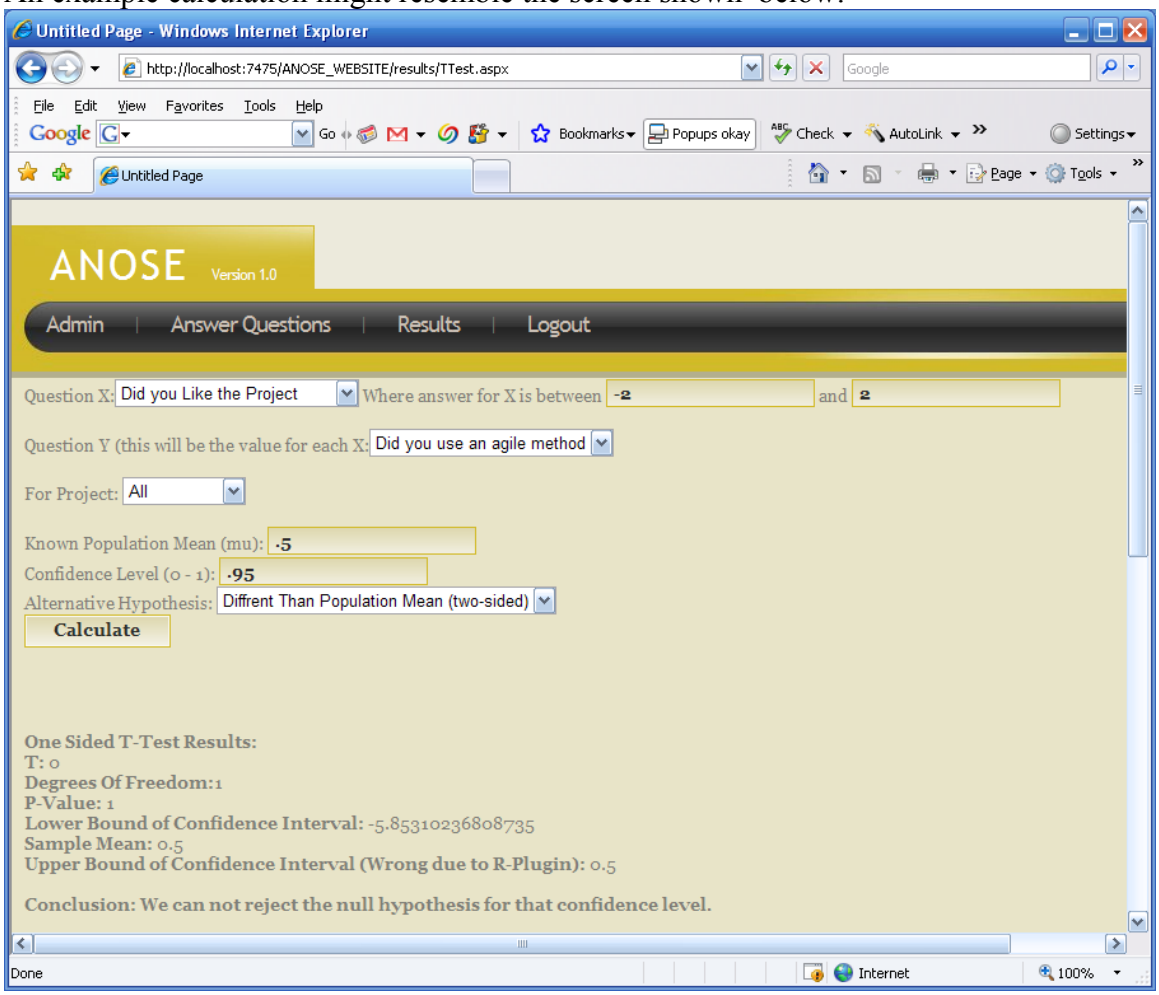
Question X is the sample from the population (an example question might be “Did you use an agile method.”). An administrator could limit the values of X if a subset of the answers to X, such as only the “Yes” answers to the former example question, was desired.

Question Y is the value of each X. Normally this would be related to budget or time. The “For Project” dropdown is another limit on the X values.

The confidence level is the confidence level that the result is reached via a type two error. Normally this is .95 or .99.

The alternative hypothesis drop down allows users to select from two-sided, left or right side tests. In most cases, a two-sided test would be used.

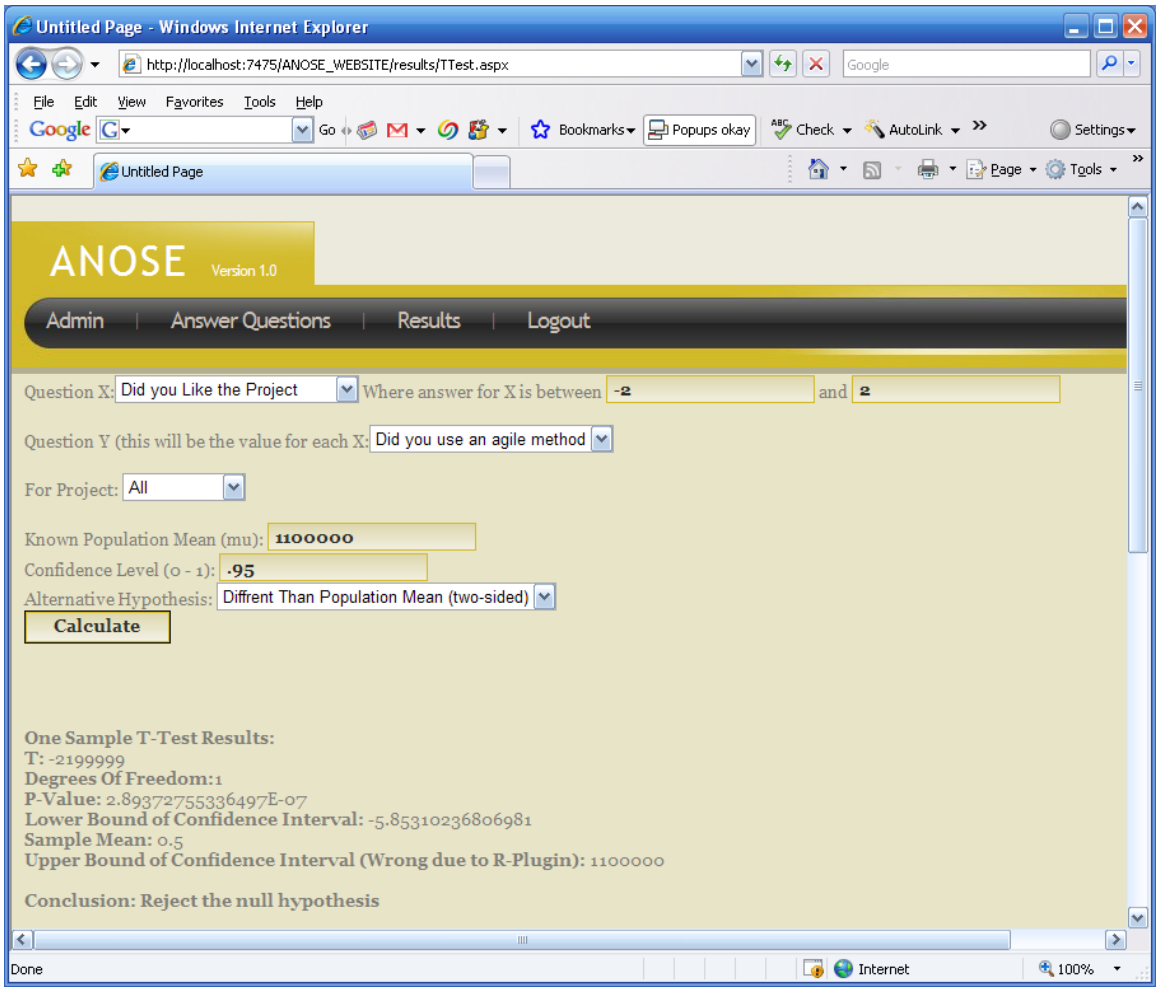
An example calculation might resemble the screen shown below:



T is the calculated T-Value. Degrees of Freedom is the degrees of freedom in this calculation. Normally more degrees of freedom is better. P-Value is the calculated P-Value. The lower and upper bounds are calculated correctly by R, but the R D-COM

corrupts them at times, so the resulting bounds can not be trusted. The sample mean is the sample mean of the values in the sample.

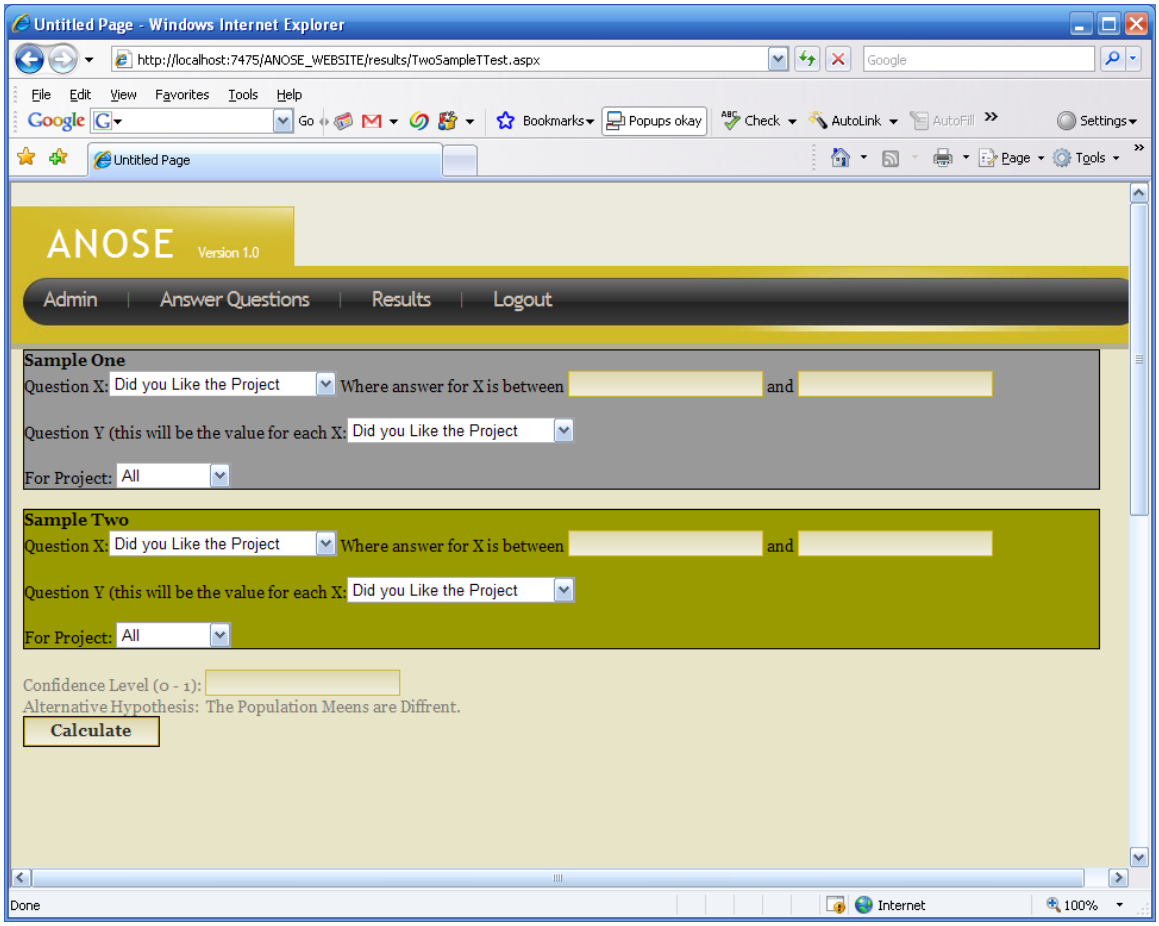
In the example above, the conclusion says "We can not reject the null hypothesis for that confidence level." It is important to note that although the null hypothesis cannot be rejected, it also cannot be assumed that it is equal.
Here is an example of a case where the null hypothesis can be rejected and it could be stated that the sample mean is statically different from the population means at the given confidence interval.



Two Sample T-Test:

The Two Sample T-Test can be used to compare two samples to determine if their means are statistically different. The Two Sample T-Test tends to be more useful than the One Sample T-Test as the population mean is usually unknown.

To perform a Two Sample T-Test, the administrator would login as an admin, go to the results menu section and select Two Sample T-Test. The resulting screen should resemble the image shown below.



This screen functions similarly to the One Sample T-Test screen, and now allows two samples to be selected, shown as two colored boxes. The administrator would enter a confidence level and click on calculate, which would display T-value, Degrees of Freedom, P-value and conclusion just like the one sample T-Test.

Technical Notes:

This section will detail notes that are useful to developers. Users of the system do not need to read the rest of this section.

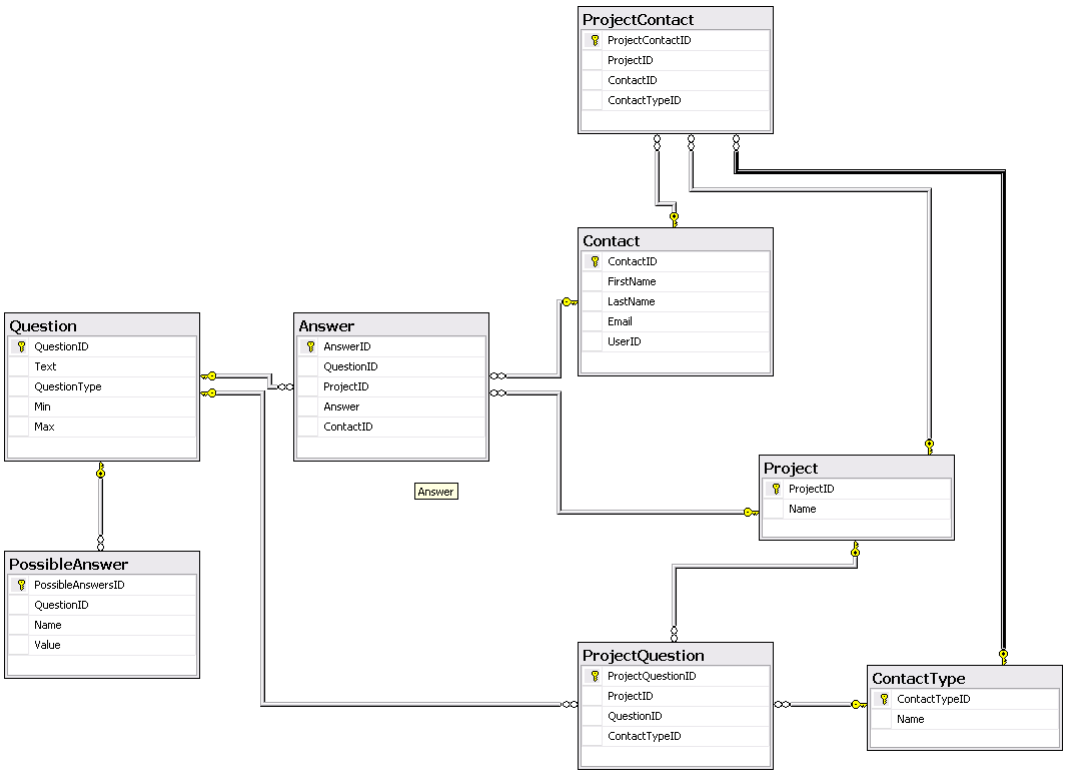
Database Design:

The database design of ANOSE is actually fairly simple. The main tables are questions, answers, user, and project. The tables are self explanatory and relations are pretty straight forward.

The PossibleAnswer table is related the question table. The PossibleAnswer table holds possible answers to drop down questions. If a question of type selection is specified, ANOSE will generate a drop down with values from the PossibleAnswer where the PossibleAnswer.QuestionID is equal to the QuestionID being displayed.

The ProjectQuestion table is another key table. This table determines who (User Type) gets asked what (Question) about what (Project). Note that if an entry exists in this table then it will be asked for that user type, question, and project. If it does not exist, then it will not be asked. The QuestionProjectODS implements an ObjectDataSource for the operations on this table in a programmatic fashion.

Bellow is a diagram of the database. Note that the ASP.Net membership tables have been left out. For more information on the membership tables see [6].



Website Design:

The website design of ANOSE is typical of an ASP.Net 2.0 project. Some programmers may not be familiar with the layout of a ASP.Net 2.0 website, so this section will provide a quick overview.

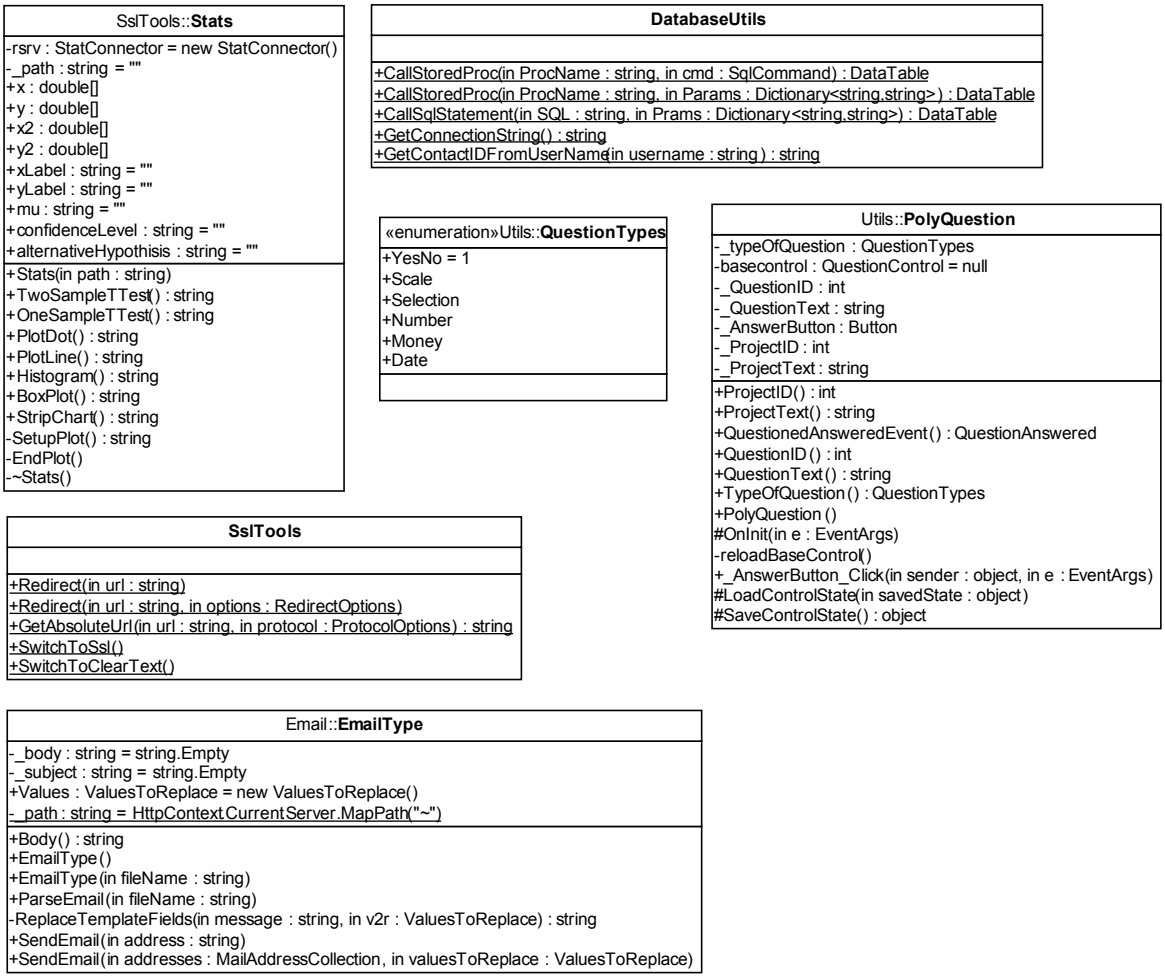
Each web page contains a code behind (the .aspx.cs file). The code behind includes the code for the operation of the page. Classes that can be accessed by all pages are located in the App_Code directory.

The one and only config file for the web application is the web.config file. The AnoseConnectionString serves as the database login and as the server information the application uses to access a database. The location tags at the bottom of the file specify which folders require login and role to access.

Each webpage has a master page and theme specified in the page directive at the top of the aspx file. The master page is ANOSEV1.master, the page html is inserted where the ContentPlaceHolder tag is. The Theme is ANOSEV1 is located in App_THEMES. Note that a skin file is not used; all CSS information is located in the CSS file.

Key Classes:

ANOSE V1.00 has hundreds of classes in it. Most of them are only used once per page (each page is a class in .Net). Below is a quick overview of some important classes.



Stats Class:

The Stat class is ANOSES wrapper to the R DCOM library. The x and y vectors are value vectors for the functions such as PlotDot() etc.

SSlTools:

SSLTools is a class to help make sure that users are using the website from a https server. This is currently unused in this version of ANOSE, but could be very useful in production.

Email Class:

The email class is a wrapper class for the .Net email classes. This is currently unused in this version of ANOSE.

PolyQuestion:

PolyQuestion class is a control that can represent all question controls. It is much easier to create one instance of the polyquestion in the aspx file then to dynamically create an instance of the correct question type. Note that due to a bug in .Net 2.0 when using the polyquestion control, the answer button must be pressed twice.

QuestionTypes:

This is an enumeration of question types that corresponds to the question type on the database. If a new question type is added to ANOSE, it will have to be added here and in polyquestion.

DatabaseUtils:

The DatabaseUtils class is a static class that contains utility functions to access the database. These functions return a data table and are much easier to use then the normal ASP.Net database functions.

Menu Control in detail:

The menu control used in ANOSE is extremely configurable and integrates well into .Net membership classes. In the menu folder there is a user control called MenuControl.ascx. The user control uses the xsl file to transform the xml file in html and JavaScript to render the menu. This means that items can be added and removed by editing the the xml file. Note that items can be removed from the xml by the user control based on the role that the current user is in. This is how the menu items are hidden from users who aren't logged in yet.

Question Controls in detail:

The source code to each question control is located in the App_Code\Questions directory. All controls inherit from a base control called Question. The Question base control already does most of the rendering and variable storage, so overriding the Question control makes it very easy to create new controls. For example here is the complete source code to the NumberControl:

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

namespace Utils
{
   public class NumberControl : Utils.QuestionControl
   {
      public NumberControl()
      {
```

```
    }

    public override bool isValid
    {
      get
      {
        double dec;
        bool result = double.TryParse(Answer.Replace("$","").Replace(",",""), out dec);
        if (result == false)
        {
          _errorMsg = "Invalid Double.";
        }

        return result;
      }
    }

  }
}
```

PolyQuestion Control in detail:

The PolyQuestion Control renders subcontrols, including validators, from a bound data field. This can be done by adding it to the page, setting the QuestionID and ProjectID. The PolyQuestion Control will generate an event (QuestionedAnsweredEvent) when a question is answered. All the rendering of the controls is handled by the PolyQuestion.

The code for the polyquestion control is located in App_Code\Questions\PolyQuestion.cs. The code is simple, but it's a fairly large class. The rendering happens in the reloadBaseControl function and uses the ViewState to dynamically select the type of control to add an event handler to the Answer button.

Note that there is a bug in the current version of the PolyQuestion Control. The answer button must be pressed twice due to a .Net 2.0 bug. The bug causes page rendering and loading of the viewstate to happen in different orders depending on what type of postback is picked by .Net. It is possible that this bug has been fixed in future versions of .Net.

Know issues in ANOSE Version 1.00:

ANOSE was developed using agile methods, and as such has a few minor bugs in this release.

- Manage Users should be on User Management screen
- Manage Projects Page says UserID where it should say user
- Manage Projects Page says UserTypeID where should say user type
- Manage Questions Page is missing the drop down value editor
- Membership Page has membership list wider then content area
- Detail button throws an error on Membership Page
- Reset Password button generates impossible passwords on membership page
- New User isn't bold on Membership page.
- Answer button must be clicked twice on answer question page.
- Plot One Dimension page shows a red x for picture on load.
- Plot Two Dimension page shows a red x for picture on load.
- One Sample T-Test crashes on bad input or empty set.
- Two Sample T-Test crashes on bad input or empty set.
- R-DCOM doesn't return correct parameters for the T-Tests

These and other bugs are tracked via the bug tracker at http://anose.sourceforge.net.

Future Plans for ANOSE:

A SourceForge GPL 3 project has been setup and development will continue to fix the above known bugs and more statistically functionality. The SourceForge project is located at http://anose.sourceforge.net.

Below are planned additional functionalities:

- Export Data to CSV
- A Better Export to R
- Output the R plot code with the plot it self
- Plot Formatting
- ANOVA Support
- Linear and Non-Linear regression
- Z-Test
- Wilcoxon Tests
- Chi-square tests
- Better printing support using CSS
- Improve the account creation process
- Improve the user type/question/project management page to make it easier to manage questions.
- Email users when they have questions to answer.

References:

[1]Panel on Statistical Methods in Software Engineering, National Research Council. Statistical Software Engineering, 1998, ISBN-13: 978-0-309-05344-0 http://www.nap.edu/catalog.php?record_id=5018

[2]Tucker, John. Statistical Software Engineering 1998 http://stinet.dtic.mil/cgi-bin/GetTRDoc?AD=ADA344440&Location=U2&doc=GetTRDoc.pdf

[3] Singpurwalla, Nozer. Wilson, Simon.
Statistical Methods in Software Engineering: Reliability and Risk
1999, ISBN-13: 978-0387988238
http://www.amazon.com/Statistical-Methods-Software-Engineering-Reliability/dp/0387988238

[4] Land, Lesley Pek Wee. Aurum, Aybüke. Handzic Meliha.
"Capturing Implicit Software Engineering Knowledge,"
aswec, p. 0108, 13th Australian Software Engineering Conference (ASWEC'01), 2001
http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/&toc=comp/proceedings/aswec/2001/1254/00/1254toc.xml&DOI=10.1109/ASWEC.2001.948504

[5] Laitenberger, Oliver. Dreyer, Horst M.
Automated Software Engineering Data Collection Activities via the World Wide Web: A Tool Development Strategy applied in the Area of Software Inspection
http://citeseer.ist.psu.edu/96533.html

[6] Mitchell, Scott.
Examining ASP.NET 2.0's Membership, Roles, and Profile - Part 1
http://aspnet.4guysfromrolla.com/articles/120705-1.aspx