WORKING PAPER NO.242

Engineering the Software Requirements of Nonprofits – A Service Learning Approach

by

Shankar Venkatagiri

January 2006

Please address all your correspondence to:

Shankar Venkatagiri
Indian Institute of Management Bangalore
Bannerghatta Road
Bangalore 560076, India
Email: shankar@iimb.ernet.in
Phone 080-26993327
Fax 080-26584050

Engineering the software requirements of nonprofits - a service-learning approach

Shankar Venkatagiri
Assistant Professor, Quantitative Methods & Information Systems
Indian Institute of Management Bangalore
(91)-80-26993327
shankar@iimb.ernet.in

ABSTRACT

This paper is a cross-study of service-learning projects executed by student groups in a 10-week course on software engineering. The principal benefits of service-learning are demonstrated by the groups in this setting. The course is structured to support the project activities; timely brainstorming and negotiation roleplay exercises help the teams arrive at pragmatic baselines with their clients. The study highlights overlaps in the software requirements of nonprofits. The paper apprises the reader of some common mistakes committed by the various stakeholders, some of which can eventually undermine the project's mission.

Categories and Subject Descriptors

D.2.1. [**Software Engineering**]: Requirements/specifications. – *elicitation methods, methodologies.*

General Terms

Management, Documentation, Experimentation

Keywords

Software engineering, service-learning, nonprofit, NGO, requirements, brainstorm, negotiation

1. INTRODUCTION

Information systems are conceived, constructed and maintained by teams of diverse stakeholders, rather than by individual programmers. Tools and techniques such as integrated development environments and model-driven architectures address the mechanical aspects of software - the *accident*, according to Fred Brooks. However, an experiential knowledge of skills like brainstorming, negotiation and team coordination is imperative to tackle the *essence*.

The software engineering domain can be regarded as the confluence of managerial and technical thinking about software, since it has at its core issues of estimation, metrics, quality, etc. Software engineering courses typically cover a good deal of theory, ending with a four-week project. Students are exposed to various methodologies, estimation theories, best practices, etc. What is largely missing is an emphasis on the human dimension, as detailed by DeMarco & Lister in their *Peopleware* classic[2]. Software professionals who fail to factor in this people angle perform poorly in critical tasks like effort estimation.

International Conference on Software Engineering '06, May 22-28, Shanghai, China.

Contrary to expectations, even business schools, which claim to train future managers, sidestep this "soft" but important aspect. There has been extensive discourse (see Mintzberg [12]) on the narrow, analytical outlook promoted by management education today. Consequently, graduating managers lack a well-integrated perspective of a workplace and are ill-equipped to deal with the people that drive the quotidian processes at their organisations. Decisions that these managers make can have an adverse impact on their stakeholders and society at large. It is arguable that these issues equally concern a software professional's training.

2. SERVICE-LEARNING

Jacoby[7] defines service-learning as follows:

> *...a form of experiential education in which students engage in activities that address human and community needs together with structured opportunities intentionally designed to promote student learning and development; service-learning combines service objectives with learning objectives with the intent that the activity change both the recipient and provider of the service...*

Students address community needs by working with nonprofits, while they are enrolled in a course with classroom interactions. Godfrey *et al*[6] propose service-learning courses as a measure to mitigate the shortcomings of a manager's education. They list the following motivations: learnings gained from a real setting, opportunities for reflection, the reciprocity of the benefits to all the stakeholders (students, organisations and the community), and sensitivity to one's civic responsibilities, which has a positive long-term impact on the manager.

A course in software engineering provides a golden opportunity to incorporate service-learning. The author has "implemented" such a course, which features term projects with NGOs. The paper studies these projects based on their submitted artifacts. The analysis also draws from informal forum discussions and more formal feedback supplied by the participants.

3. THE NGO SCENARIO

Several forces influence the sustainability of nonprofit efforts. NGOs are evaluated on the efficiency and transparency of their resource usage, the reach and scale of their beneficiaries and the impact that they have upon society. Electronic networks play a critical role in responding to each of these challenges and may even lead to innovations in the entire NGO sector. While many nonprofits are able to access the Internet and avail of basic email facilities, a majority of them are yet to employ IT for strategic functions like forecasting and policy-making[1].

A fairly representative survey of American nonprofits[5] reveals key inadequacies in technology planning and implementation: absence of an MIS department, lack of staff competencies to

undertake customization, inability of the packages to address the organizational context (e.g. country-specific accounting rules), low technology budgets, etc. Even when these complications are absent, a survey of nonprofit accounting software packages[8] illustrates the complexities involved with package selection. NGOs are thus forced to operate with tools that are limiting – many of them maintain their data in spreadsheets, or worse, in a plain text document. Furthermore, artificial processes are constructed around these artifacts, complicating their usage.

Students of software engineering can resolve the above predicament by helping nonprofits formulate their IT issues via detailed requirements exercises. Equipped with a proper set of artifacts, a nonprofit will be in a better position to seek funding. IT vendors who are then contracted to implement a solution would readily appreciate the convenience of such artifacts. Alternatively, it is conceivable that open source communities can use these artifacts to arrive at a solution.

4. COURSE CONTEXT

The Indian Institute of Management Bangalore (IIMB) conducts a full-length program titled PGSEM (post-graduate program in software enterprise management) targeted at IT professionals. Besides imparting general management education, it offers courses immediately relevant to the software practitioner, such as software project management (SPM) software engineering management (SEM), systems analysis & design, etc.

SEM is offered as a follow-up to SPM, which covers topics like planning, estimation, scheduling and configuration management. The author has evolved the SEM course format over three years. This 10-week, 30-hour course is a mix of instructor-led sessions, student presentations, debates and controlled team exercises. Initial attempts to allow the groups to devise their own problems resulted in applications that were contrived. Hence, the author decided to solicit realistic problems (e.g. from local NGOs) and encourage the students to choose from among them.

4.1 Project methodology

The use case driven process laid forth by Leffingwell & Widrig in *Managing Software Requirements*[9] served as a guideline for the conduct of the project. The instructor supplied a "template project site", which detailed the milestones (see Appendix A) on separate web pages. The site also included a presentation on the overall process to be followed by the group. This helped the students to tackle process-oriented questions raised by clients. Some milestones (e.g. feature elicitation) required the students to visit and interact with their clients.

4.2 Learning management system

The instructor employed Moodle, an open source learning management system (LMS) to communicate with students outside the class. Moodle is freely downloadable (moodle.org), easy to deploy and operate on a LAMP configuration. The LMS supports standard features such as forums, document uploads, grade distribution, event calendar, etc.

5. PROJECT INCEPTION

The NGOs in Bangalore are indexed by several online databases, one of which is Fingertips (fingertips.sutradharindia.org). This database categorizes NGOs into child development, education, health, disability, recreation, support and helplines. Each NGO is listed with a detailed profile that includes contact information.

5.1 Initiation

Prior to the start of the course, the author contacted several local NGOs to discuss the possibility of student groups spending a term with them. It was important to explain the project's mission to the administrators in non-technical terms. Many perceive software as shrink-wrapped products that can be purchased and installed. Indeed, the software that they work with on a daily basis are of the COTS variety (e.g. an Office suite) and hence the misconception. Most of the NGOs owned computers, so the basic hardware requirement was met. Some also had Internet access provided by a low-cost ISP.

While it was important for an organization to buy into the effort, it was essential that they understood *how* they were going to benefit from their participation. The author explained that at the end of the term, they were going to receive the equivalents of a blueprint (SRS) and a scale model (prototype) of a solution to their stated problem. A formal letter was then sent to them, urging them to brainstorm and paraphrase their problems for the sake of the students. Suffice it to say that only one of every four organisations contacted agreed to participate.

5.2 Selection

Appendix B lists the NGOs that sponsored between one and four projects. The instructor posted their problem descriptions via the LMS on a dedicated forum, requesting the students to select from among them. The visibility of their online responses ensured that there were no collisions in the choice. Some groups independently visited nonprofits that they were familiar with to seek out their problems. They posted them on the LMS indicating additional opportunities to their peers.

5.3 Motivation

Initially, the groups were lukewarm to the idea of working with nonprofits, mostly because of misconceptions of inefficiency. However, once they met the administrative heads as well as their ultimate stakeholders (blind students, poor childen, marginalized communities, etc.), they became sensitised to the problems tackled by their NGOs. Administrators were working full-time, some of them after quitting lucrative careers in law, IT, etc. The groups realized how software solutions could greatly impact the day to day functioning within these organizations.

5.4 Coordination

An effort of this scale warrants intensive coordination. The instructor created three "internal projects":

- ***Project Gallery:*** Provisioning space for the projects and taking them online with a web server.
- ***Project GetGo:*** Supplying working code for database connectivity, report generation, etc.
- ***Project Reach:*** Polling the NGOs with surveys to facilitate their feedback on the project artifacts.

6. INSTRUCTIONAL SUPPORT

In addition to the textbook[9], the students had to consult readings from practitioner journals like IEEE Software, CACM, Harvard Business Review, etc. The instructor scheduled the course activities well in advance of their potential application. Role-play exercises were conceived around brainstorming and negotiation, in order to prepare the groups for feature elicitation activities with their clients.

The instructor elaborated on the three phases of brainstorming, viz. ideation, reduction and prioritization, and the students were assigned a chapter reading on facilitation from the classic by Doyle & Straus[3]. Being IT professionals – a breed notorious for its inability to decline any client request – the students had to be educated on negotiations so they could arrive at a realistic baseline of requirements. The instructor covered the principled negotiation framework of Fisher and Ury [4], after assigning a complementary reading by Sebenius [13].

6.1 Brainstorm

The first role-play exercise involved facilitating a brainstorming session with groups of seven. Their interactions were captured on video by the instructor and two assistants. The topic was of burning significance: the issue of caste-based job reservations had just then been raised by India's prime minister. The groups were asked to brainstorm on how a ruling had to be statutorily accommodated by their workplace. Facilitators were appointed based on their prior interactions with the instructor. Each group member was assigned a role to play, and required to exhibit a problem behavior (see Table 1).

Table 1: Problem behaviors

Role	*Behavior*
Back-seat driver	Provides unsolicited advice to everyone
Broken record	Keeps bringing up an idea repeatedly
Critic	Puts down the ideas generated by peers
Dropout	Switches off and looks elsewhere
Head shaker	Vigorously shakes his head at an idea
Loudmouth	Speaks loudly and ignores everyone
Whisperer	Continuously whispers to the neighbor
Wildcard	Throws out wild, offensive ideas

Courtesy Doyle & Straus [3]

Although many students had declared that they were familiar with brainstorms (IT professionals hold free-form discussions that are mistakenly labeled so), their performance on this controlled exercise left a lot to be desired. A ready benchmark was the number of ideas raised during the ideation period of 20 minutes. Groups that did their homework on the assigned reading performed well (20-45 ideas), while unprepared ones only managed 8-10 ideas.

The problem behaviors caught most facilitators by surprise. They chose to ignore the "troublemakers" and continue despite the interruptions. Doyle & Straus underscore the importance of setting ground rules at the beginning of meetings. Facilitators who established and enforced them succeeded in tackling most problem behaviours. The idea reduction phase went smoothly with such groups, while the prioritization phase was meaningful only to groups that had chalked up a significant number of ideas.

The instructor went through the video footage in class, exposing good and bad facilitation practices. He outlined how he would have facilitated a brainstorm. The feedback from the class on this exercise was quite positive. One student even had the opportunity to conduct a brainstorm at his organization the coming week and received rave praise from his colleagues.

6.2 Negotiation

The next exercise involved a classic contract negotiation by two 5-member teams, one representing an overseas financial firm wanting to outsource a project, and the other representing an Indian IT consulting firm bidding for the contract. Each side was supplied with confidential information pertaining to their specific context. For instance, the client team had an idea of their reservation price, having met with four other IT firms. They also had to meet an impending SEC deadline, which pressured them into seeking the solution. The consultants were aware that several of their colleagues were on the bench.

While the two groups negotiated at the "stage", the remaining class was divided into two columns to support either the client or the consultants. These sections were also supplied with specific instructions relating to their context. A "messenger" conveyed their suggestions efficiently to their negotiating team. Periodically, team members could opt out of the negotiations and a replacement would be sent from the audience.

One benchmark was how close to the client's reservation price the consultants reached during the course of the negotiations. Conducting this exercise across two sections, it was evident that groups adopting a principled negotiation approach arrived at an efficient resolution to the problem, while acting as a joint team. Free-style negotiators fell into the trap of positional bargaining and failed to elicit sufficient information in order to make a justifiable bid at the end.

After the exercise, the instructor asked the camps to exchange their confidential instructions, which lead to a heightened appreciation of the principled approach. The feedback on this exercise was also positive. Students delved deeper into concepts like BATNA using the forum discussions on the LMS.

7. CROSS-PROJECT ANALYSIS

Table 2 highlights some overlaps in the features requested by the participating NGOs, which were usually peculiar to their sector. For instance, training-oriented organizations desired a placement database for their graduates.

Table 2: Commonalities in features

Feature	*Nonprofits*
Web presence	CRT, Gerizim
Program monitoring	Akshara, Hippocampus, SKIP
Beneficiary profile	APD, Dream-a-Dream, Gerizim, Margadarshi
E-payment gateway	APD, Gerizim
Patient history	Haemophilia Society, Margadarshi, YRG Care
Placement database	APD, SKIP
Library management	Mitra Jyothi, Samarthanam Trust

While most nonprofits wished to establish a web presence or IT-enable their existing processes, one NGO (Hippocampus) harnessed the group to devise a mathematical model for program assessment. IT-savvy NGOs voiced common technical needs such as session management, secure access, web-centric architecture, (free) open source platforms, etc.

7.1 Progress

Successful groups were able to generate concrete examples of the various constructs discussed in the classroom. They drew up problem statements based on different stakeholder perspectives. They conducted detailed interviews with principal stakeholders along the lines of the template in Leffingwell & Widrig[9]. Most groups distinguished between stakeholder needs, application features and system requirements. Equipped with the learnings from the mock classroom exercises, they facilitated brainstorms and negotiated a prioritised feature list using a voting process.

7.2 Deliverables

Overall, 15 projects (< 50%!) created satisfactory deliverables. On a positive note, although the instructor had emphasised that the prototype would be treated on par with any other milestone, eight of the submissions contained impressive screen shots, complete with an intuitive navigation. This indicated their empathy with the client's problem. One group that worked on a micro-finance application felt that there was strong commercial potential for their application if it were fully developed.

On a negative note, many deliverables contained basic mistakes. Ambiguous feature specifications were commonplace. For example, the term "data" was ill-qualified: *"Enable data entry"* is a badly specified feature. Use case names and action steps had GUI details embedded in them. Technical requirements such as a secure login received undue attention. Some projects proposed requirements that did not fully satisfy their stakeholder needs. Some groups reinvented the wheel instead of customizing software that was readily available in the public domain.

7.3 The people dimension

The groups described their overall experiences in a final report. Suffice it to say that these experiences can never be simulated satisfactorily in a classroom setting.

Steve McConnell has coined the term "fuzzy front-end" [11] to describe the period of indecision before the start of a project. Many groups experienced this first hand, with clients who were uncertain about their IT issues and failed to perform the requisite groundwork. Some clients were simply unavailable for discussions. Consequently, the associated groups requested their deadlines to be extended. However, they were unable to make satisfactory progress and had to submit incomplete deliverables.

Although clients were promised only a "blueprint" prototype, some were satisfied by nothing less than a fully-functioning application. Tackling this unanticipated scope creep taught the groups some valuable lessons about expectation management.

Some groups engaged with their clients in innovative ways: they fixed virus-infected machines, enabled spreadsheet macros, etc. Once the clients discovered the commitment and competence of these students, they were more forthcoming with their problems.

Groups that interacted with the ultimate stakeholders were enlightened by the experience. They came to empathise with the harsh realities of visually impaired children. They were impressed by the zeal of the volunteers, who were paid poorly. The reach of the NGOs amazed students who otherwise dealt with clients of "lesser" social impact at their workplace. For example, NESA targets over five million marginalized people in 6,000 villages. This increased the team's motivation to solve their problem, as was evidenced by their thorough prototype.

7.4 Stakeholder impact

Students were forced to transition from an implementation mindset (as IT professionals) to an inquiry mindset. Groups that worked with NGOs that sponsored multiple projects had to carefully eliminate overlaps as they met with different teams. Good time management was critical, since the groups had to synchronize the availability of their clients with the project deadlines. Clients that were not IT savvy came to embrace technology tools as all project deliverables were posted online.

The course feedback was quite positive, although some students complained of the excessive burden it placed on their schedule. When asked if they would continue working with their clients, the response was overwhelmingly positive.

7.5 Instructor lessons

The instructor recognised the need to assert the level of support from the client NGOs. The projects failed to make sufficient inroads in instances where the administrators did not contribute fully to the activities. On the other hand, students with a precise idea of their project goals and their community impact tended to perform better. These findings resonate well with recent service-learning literature (for example, see Lester *et al* [10]). For the next iteration, efforts shall be made to meticulously define the projects *a priori* so as to give students an early start, as well as to weed out any disinterested clients.

8. CONCLUSION

This paper illustrates a service-learning pedagogy realised by a software engineering course. The course succeeded in providing the students with a real setting, reciprocal stakeholder benefits and a heightened sensitivity to their social responsibilities. Future service-learning efforts can distill the overlaps out of the separately created system requirements into a versatile set of artifacts that cater to an entire sector. This shall provide intense reflective opportunities for the students.

9. ACKNOWLEDGMENTS

The author deeply acknowledges the PGSEM class of 2006 for their spirited participation in the projects. Krishanu Rakshit and Nitin Chokhani served as capable teaching assistants, doubling as videographers when the situation demanded. Finally, the author thanks the nonprofits that sponsored the projects and provided the students with valuable insights.

10. REFERENCES

[1] Burt, E. and Taylor, J.A. Information and communication technologies – reshaping voluntary organizations? *Nonprofit Management and Leadership*, Volume 1, No.2, Winter 2000, pp. 131-143

[2] DeMarco, T. and Lister, T. *Peopleware: Productive Projects and Teams.* Dorset House, 2nd Edition, 1999.

[3] Doyle, M. and Straus, D. *How to make meetings work: the new Interaction method.* Berkley Publishing Group, 1993.

[4] Fisher, R., Ury, W.L. and Patton, B. *Getting to Yes – negotating agreement without giving in.* Penguin, 2nd/Reprint Edition, 1991.

[5] Forster, J.J. and Outon, P.M. Southwestern Pennsylvania Nonprofit Technology Survey 2004. Bayer Center for Nonprofit Management, Robert Morris University.

[6] Godfrey, P.C., Illes, L.M. And Berry G.R. Creating breadth in business education through service learning. *Academy of Management Learning & Education*, Volume 4, No. 3 (Sept. 2005) pp. 309-323

[7] Jacoby, B. Service-learning in today's higher education. In B. Jacoby (Ed.), *Service-learning in higher education.* Jossey-Bass, 1996.

[8] Jones. R.A. Sizing up NPO software. *Journal of Accountancy* (Nov. 2000) pp.28-44

[9] Leffingwell, D. and Widrig, D. *Managing Software Requirements: a use case approach*. Addison-Wesley Professional, 2nd Edition, 2003.

[10] Lester, S.W., Tomkovick, C., Wells, T. and Flunker, L. Does service-learning add value? Examining the perspectives of multiple stakeholders. A*cademy of Management Learning & Education*, Volume 4, No. 3 (Sept. 2005) pp. 278-294

[11] McConnell, S. Best Practices. *IEEE Software* Volume 13 No.5 (September 1996)

[12] Mintzberg, H. Managers not MBAs: a hard look at the soft practice of managing and management development. Berrett-Koehler, 2004.

[13] Sebenius, J. Six habits of merely effective negotiators. *Harvard Business Review, (*April 2001)

Appendix A: Project milestones (* indicates that a client visit is required)

Milestone	*Artifact(s)*	*Deadline*
Discovery*	1. Root cause analysis, stakeholder identification and solution constraints	Week 3
Problem formulation	2. Problem statements from different stakeholder perspectives. 3. (Optional) Problem frames construction	Week 4
Needs analysis*	4. Interviews with the stakeholders – follow the template in Leffingwell & Widrig[9]. 5. List of needs expressed by the clients	Week 4
Feature elicitation*	The group facilitates a brainstorm with the client on the features for the proposed system. 6. A description of the brainstorm together with a list of prioritized features.	Week 5
System definition*	7. List of actors, brief descriptions of use cases and a use case model depicting actors and use cases. 8. Vision document according to the template in Leffingwell & Widrig[9].	Week 6
Scope management	The group is asked to double as a development team and guess the effort and risk for each feature. The clients supply risk information from their perspective. The prioritization is then finalised via email. 9. Categorized priority-effort-risk table of the features, two baselines based on the PER table.	Week 7
Requirements refinement	10. Refined use case model, based on a what-if analysis of the initial use cases. 11. SRS document based on the template in Leffingwell & Widrig[9]. 11. Exercise to weed out ambiguity in specification.	Week 8
Verification	12. Traceability pyramid of requirements to features to needs to patch up any "holes".	Week 8
Test plan	13. Test cases for the first baseline, created from use cases detailed in Leffingwell & Widrig[9].	Week 9
QA	14. Checklist of items that track the whole process, supplied by Leffingwell & Widrig[9].	Week 10
Prototype*	15. Sample screens must be submitted with a write-up on their functionality.	Week 10
Project report	16. Lessons learned by the term, in a medium sized write-up.	Week 10

Appendix B: Participating NGOs and the sponsored projects

NGO	*Categories*	*Projects and their key features*
Abilities- home for the blind	Disability	***ScRead*** – screen reader for the visually impaired
Akshara Foundation	Children Education	◆ ***Outreach*** – Statistical analysis and reporting of program effectiveness with data on children, teachers and trainers. ◆ ***CAMP*** – Self-paced mathematics tutorial program based on proprietary methods, supporting analysis of test data. ◆ ***MIS for In-School Learn-to-Read program*** – Reporting tool to compare performance at student, class, school and block levels. ◆ ***MIS for Pre-School program*** - Reporting tool to compare performance at student and school levels, profile database for children, parents, volunteers.

NGO	*Categories*	*Projects and their key features*
Association of People with Disability (APD)	Children Disability Training	◆ ***Communication and fund-raising*** – Donor and sponsored child database, activity reporting, mid-day meal scheduling ◆ ***E-payment gateway*** - for donations and purchase of in-house products, database to display trainee resumes
Association for Promotion of Social Action (APSA)	Self-help groups	***Self-Help Groups Automation*** – MIS to maintain self-help group profiles and micro-finance loan transactions, consolidation reports
Child Rights Trust (CRT)	Children Training	◆ ***CRTWeb*** – Bulletin board for trainees, mechanism to report child labor incidents, organizational information dissemination ◆ ***TrnAll*** - Training scheduling and content management, feedback collection and report generation at village and town levels ◆ ***Adhikaar*** - Data collection and historical analysis tool to support development initiatives.
Dream a Dream Foundation	Children	◆ ***Beneficiary management system*** – Storage and reporting of beneficiary information, management of various project initiatives. ◆ ***Volunteer management system*** – Storage and filtering of volunteer profiles, communication with volunteers.
Eco-watch	Community	***Ecosys*** - Membership database, mass mails to generate awareness
Gerizim	Children	***Web site*** – Information dissemination, E-payment gateway for donations, reminders to donors, beneficiary look-up
Haemophilia Society	Health	◆ ***H-Soc Member Information System*** – Database to maintain membership details and treatment history ◆ ***HSoc-Acc*** - Medicine purchase and sales registry, inventory management
Hippocampus Reading Foundation	Children	***Hippo*** - Mathematical modeling of benchmarks for a reading program, statistical reporting at the level of class, school, librarian.
Janaagraha	Community	***Jasmine*** – Portal for three modules of existing information management system, volunteer registration, data entry for the modules.
Margadarshi	Disability	***Margadarshi Information System*** – Database of surgeries performed, patient tracking, donor contribution management, etc.
Mitra Jyothi	Disability	***Mindows*** - Library management system, membership, subscription functions
Mobility India	Disability	***MIST*** - Survey creation and aggregation for R&D proposals, repository for R&D artifacts, enabler for outsourcing product parts
New Entity for Social Action (NESA)	Community	◆ ***NESAILS*** – Program survey creation, performance reporting ◆ ***NESAMIS*** – Program profile database, report generation
Samarthanam Trust	Disabled	◆ ***Library management system*** – electronic document management ◆ ***Sparsh*** – Donor and volunteer database, event alerts
Skills for Progress (SKIP)	Training	◆ ***Electronic Resource Center*** - Placement database, donor fund utilization reporting, course announcements ◆ ***SKIP123*** – Newsletter mailing, member organization profile database, program statistics, financial transaction repository
SOS Children's Village	Children	Database revamp and upgrade
Technology Informatics Design Endeavour (TIDE)	Rural	◆ ***Knowledge Management System*** – Document workflow and search engine, geographical categorization ◆ ***Financial Management System*** – Tax report generation, tax calculations, project fund forecasting
YRG Care	AIDS	◆ ***Datebook*** - Appointment scheduling, patient report browsing and download, calendaring, activity reports ◆ ***Patient Tracking System*** – Barcoding to track patient in and out times, integration with existing hospital management system.