

CS 632 IT Project Management

Web Project

Description of Sample Deliverables and Documentation Expected in your Final Project

Web Project	1
Description of Sample Deliverables and Documentation Expected in your Final Project	1
Project Management Deliverables	3
Team Blogger Account	
Project Plan	3
Templates	5
Weekly Status Reporting	6
Web Team Roles	8
The Producer	10
Appendix A: Technical Specifications	17
Appendix B: Software Project Management Documentation Examples	32
The Change Management Process	32
Quality Planning	34
Milestone Commitments	35
Technical Architecture	36
Strategy for Meeting Quality Goals	37
Status Reporting	39
Appendix C: Closure Report	40

Project Management Deliverables

Team Blogger Account

Your next job is to create a Team Blogspot page.

1) **Team BLOGSPOT**

Create a Blogspot for you team. This will be your open **project notebook**.

This notebook will be used to document project progress, individual contribution, issues problems etc. Your instructors Facilitators at Large, Prof. Marcus Goncalves and I will aggregate your team blogspots using RSSOwl (<http://www.rssowl.org/>) and read the various blogspot accounts from all the 30+ teams on a weekly basis and guide you. (Notice the timely use of RSS technology to manage 30 teams at the same time! The other alternative is to go into 30 websites!)

2) **What about the Individual Blogspot page created in Homework 1?**

This will be used for Research *Paper* component only. In the individual blogspot which you created for HW1, please continue to post your progress. Example: List research topics that you are interested in, or post your research abstract. Your instructor will visit your blogspot page and comment on your research paper topic and guide you. Other facilitator-at-large will also visit your blogspot and post a comment on your research topic.

Finally by the fourth week you can podcast your research paper. Use your individual blogspot page to record your progress as it pertains to the research paper and issues pertaining to podcasting your research paper.

3) **Where we are going next with blogging?** (Don't worry about this now please!)

- As a project team you will publish a comprehensive set of feeds that may be aggregated by your consumers eg., publish an OPML file which RSSOwl or related software can aggregate. Sample examples of readers are:
 - IE - RSSOwl
 - FIREFOX - SAGE
 - MAC - SAGE/NEWSGATOR
- We will download iTunes and subscribe to podcast feeds.
- We will be giving you step by step Video demos showing you how to do the above. So don't panic. We have a team of industry experts to help you out.

Project Plan

Use may use the widely used PMBOK standard to guide you for the project management processes and life cycle. (Hint: Your mid-term exam has a question dealing with the PMBOK life cycle as well.)

Enter the following URL to see a sample project plan that takes you through the PMBOK life cycle.

<http://office.microsoft.com/en-us/templates/TC010774551033.aspx>

You can use it as a foundation for your project plan. Feel free to adapt this to fit the systems development life cycle as well.

Templates

In the Schwalbe textbook in APPENDIX D—TEMPLATES, you will find many templates that can be used to manage a project. Using these templates will save you a lot of time as you prepare various software project management documents. The templates are listed in the order in which you might use them and are based on the PMPBOK methodology. **We expect you to use submit several templates for each of the Initiating, planning, executing, monitoring and controlling, and closing stages that you find relevant.**

Weekly Status Reporting

The project team is required to submit weekly status reports to the instructor along with the minutes of any meetings and risks.

1. Status reports

Status reports should have the following sections.

- The executive overview section summarizes the project or subproject for the reader who is not involved with the details of the project. This section should be free of technical jargon and written in laymen terms. If you include an unresolved problem or issue in this section, you are asking the reader to get involved.
- There should be a section that provides a high-level overview of the tasks completed by each team. This can be a bulleted list by team member. This provides general information to document *who is doing what*.
- The specific progress made from the previous report should be listed. Often, this is a list of completed deliverables and resolved issues from the planned work section of the previous report.
- List the assignments, milestones, and planned work that will happen before the next report is created.
- You should provide a summary section identifying the total quantity of issues outstanding, new or opened, and closed. This summary can be helpful in many ways to the project management team in determining issue resolution and follow-up.
- You should list detail information on important issues, and issues where the resolution date is past due.
- You should provide a summary section,

2. Meeting Minutes

Meeting minutes are a way to formalize and document the decisions, issues, action items, responsibilities, and progress of the project. They are also a great way to communicate between project teams, and your instructors. Each meeting should assign team member as a scribe in rotation to take notes. The following items should be recorded:

- Note the date, start and end times
- List who called the meeting, who led it, and who was the note taker.
- List the attendees, as well as those scheduled to attend and who did not and the distribution for the minutes. Record the regrets.
- Attach a copy of the agenda and any handouts.
- Report issues and discussion; assign a unique item number to each minute for easy reference during subsequent meetings
- Report any agreements, action items, due dates, and who is responsible for future progress

3. Identifying Risks

Risks are red flags and a warning that the project would have trouble meeting its goals. One, two, or several of these warnings might not prevent the project from succeeding, but too many of these problems indicate a higher-than-normal degree of risk for the project. Consider objectively testing your project for these warning signals every few months or when major milestones are delivered.

These warnings can be grouped in three ways:

1. User/sponsor/instructor red flags
2. Communications red flags
3. Project management red flags

User/sponsor/instructor red flags indicate the degree of interest and involvement of the end users or their project team representatives. Ultimately, each successful project will transfer ownership of the new software and business processes to the end users.

Communications red flags show the openness and honesty of the project. Good communications inspires trust and acceptance of the new software and business processes. An effective communications plan will eliminate surprises and facilitate the transition to the new systems.

Look for these red flags:

- Project staff and users are not able to connect virtually to discuss day-to-day requirements, configuration, process, and design issues.
- There are communication gaps.
- The scope of the project is not clear.
- All participants (project manager, sponsor, DBA, programmer, consultant, and so forth) are not defined clearly at the beginning of the project.

Project management red flags show the degree of control and organization of the project. A well-run package software implementation project will have several distinct phases, including plan, analysis, design, build and test, transition, and production. These red flags might appear at any time:

- No formal definition of requirements has been made for the business.
- There is no provision in the work plan for development of conversion programs, security systems, testing, revisions, project management time, and so forth.
- There is excessive overtime on the project.
- The project manager has no to-do or issues list for the project. The manager does not have a work plan in sufficient detail.
- The project manager cannot give a definitive answer when asked what is done, what is not done, and which tasks are assigned to which team members. There has been no sign-off on the plan, functional requirements, design, conversion results, or system test.

Web Team Roles

You are required to submit Team Roles and Milestone targets for each role/individual as illustrated in this section

Web Team Roles for the Web Project¹

Being a Web project manager means that you are coaching a diverse team of cross-functional experts whose talents range from banner buying to data modeling. Success depends not only on the project manager's familiarity with each team member's deliverable but also on the ability of this diverse group to work together. Leading a disparate band of talented, well-paid, and sometimes cranky experts into the white heat of a large-scale, expensive, and difficult development project is no picnic. The team will look to the project manager for leadership, solace, inspiration, and days off.

The project manager needs to understand everyone's job function, the contribution their deliverable makes to the project, what inspires them to perform, and their individual quirks and idiosyncrasies. The Web project manager needs to understand his or her team on the professional, personal, and cultural level¹ in order to be the most effective coach and manager possible.

Common Web Team Roles

The composition of most development teams is a chance occurrence, usually determined by available staff, and you will rarely, if ever, be allowed the luxury of picking and choosing the people for the team. The project kickoff meeting will be the first time the team is assembled and your first opportunity to gauge the dynamics of this new amalgamation of talent. The internal dynamics of Web teams can be staggering in their complexity and scope. Managing the team, let alone the project itself, is an enormous task requiring energy and interpersonal skills.

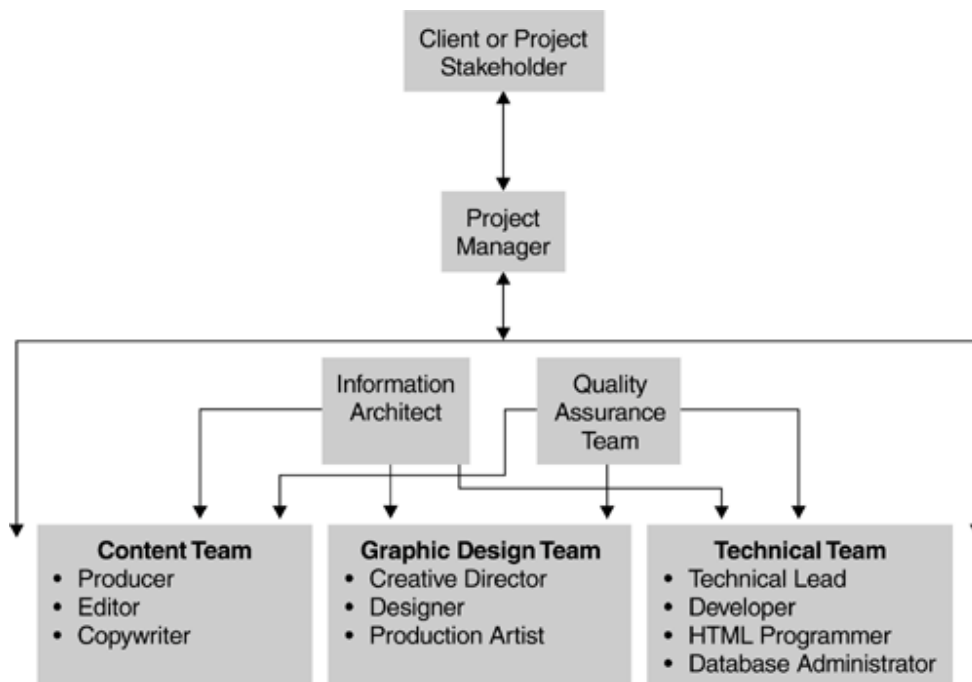
Before you can manage a Web development team you have to know the players. It's important to note that team composition is different from organization to organization, project to project, and process to process. Your Web development team may not exactly mirror the type that is described in this chapter (see the diagram below), but there will undoubtedly be many similarities. The basic roles on a typical Web development team remain relatively constant and typically include the following.

- Project stakeholder (also client or business owner)
- Project manager
- Producer
- Editor/copywriter

¹ From: Real Web Project Management: Case Studies and Best Practices from the Trenches, Thomas J. Shelford, Gregory A. Remillard, Addison Wesley Professional, 2002.

- Information architect
- Graphic designer
- HTML developer
- Developer
- Tech lead
- Database administrator
- Quality assurance engineer

Diagram of a Web Development Team



The typical Web project team is divided into three distinct groups: content, graphic design, and technology. The project manager manages across all of these groups and manages the communication between the client or project stakeholder and the team. The stakeholder normally does not play a daily, hands-on role in the development of the project but is the person responsible for initiating the project, getting the budget allocated, and wresting free the necessary resources. The project manager is the conduit of communication between the stakeholder and the team. Keeping the stakeholder or client abreast of progress on the project is one of the project manager's primary tasks.

The roles associated with the technology side of the project include tech lead, DBA, developer, and HTML developer. The roles associated with the content side of the project are producer, editor, and copywriter (who could also be the producer). The graphic design team consists of the creative director, designers, and production artists. Two roles that cross over between both the tech and content sides of the project are the information architect and the quality assurance engineer. The IA works closely with the developers on

the site architecture and with the design and content teams on ensuring that the interfaces meet usability requirements. QA engineers are responsible for testing all the components of the product from a user perspective and generally look for both functionality and display flaws and bugs.

The Producer

The Web producer has myriad tasks and responsibilities to manage during the course of a project. The Web producer role is approached and interpreted differently and morphs from company to company and department to department. It's also important to note that the producer role may also exist on the client side in the form of a product manager. These are the typical deliverables and responsibilities associated with the producer.

- Project concept/idea
- Creative brief
- Page maps
- Site map
- Final specifications
- Project timeline
- Budget
- Design direction
- Editorial content/direction
- Editorial resource management
- Third-party content deals

Traditionally the producer is much closer to the content and display aspects of the project than the project manager. The producer tends to maintain the point of view of an end user or client when working on a project. Projects may be initiated by a stakeholder, but the producer gives the project its special flavor.

The Information Architect

One of the most challenging and interesting roles on a Web development team is that of the *information architect*, or IA. The IA is the person who ensures that the Web site will be usable by human beings and ensures that the underlying structure of the Web site, including the design, content, and technology, will make sense to users. IAs come from either a technology or design background and are conversant in the finer points of both. IAs are usability experts and have logged many hours observing people interact with various types of graphical interfaces, computer hardware and software, and other objects that require quick comprehension by humans to be used successfully.

Depending on the company, IAs can wield a great deal of power on a Web initiative. They often take part in every aspect of the project build but especially in the early design and functionality planning stages. Having an IA on board helps all groups in the build process by providing a person solely dedicated to safeguarding against bad design or whacky functionality that will eventually be deemed unusable by the intended audience.

Information Architect

An information architect is an individual whose primary responsibility on a Web development team is to organize the Web site content or information into an intuitive, easy-to-understand structure from a visual, navigational, and technical perspective.

The Graphic Designer

Designers breathe life into a Web site. Typography, photography, iconography, color palettes, graphics, animation—these are the tools the designer uses to establish concept, expression, message, tone, feel, and quality. Usually the first stop for a producer working on an idea or concept, the designer provides an excellent sounding board and translates crude page mockups into beautifully polished works of art. Designers are also adept at working out thorny navigation problems that crop up in the early stages of development. While normally not bona fide information architects, good designers have an excellent grasp of how users approach Web sites and will design with the user experience foremost in mind. Designers often collaborate closely with the information architect on the page maps, site map, graphical interfaces, and navigation design.

Designers have higher profiles that add responsibilities and pressures other team members are spared. Because the graphic design and page layouts are the first tangible manifestations of the project, the designer is often trotted out before the stakeholder or client, page printouts in hand, to explain why fuchsia is the "right" color for the navigation bars and what the current standard is in "cool" Web design. Good with industry buzzwords, a designer can help sell a concept or influence the client.

Designers are long on creativity and talent but traditionally short on programming skills, but that is becoming more the exception to the rule as designers realize the value of technical knowledge. Depending on experience, they can be conversant in Web technology and understand the various components that comprise the back end of a Web site. They usually have basic HTML skills and can mock up Web pages for the browser or test how a design will translate to the screen. During the build phase the designer will work closely with the HTML resource as the final page designs are officially rendered for the screen.

Because Web design is a collaborative process, designers are good team players and understand the art of compromise. Internal clients consist of producers and stakeholders, and when working in an agency environment, they will work closely with the producer or account manager to realize the client's vision.

While being good team players and giving 100 percent on every project, designers often miss milestones and can add time to a project in their pursuit of perfection. Hovering over or badgering the designer (or any resource) as they work is not a good tactic for getting results. If a designer is consistently missing milestones and delaying projects, rather than risking an unpleasant and nonproductive confrontation, talk to the creative director about it and let him or her handle the problem. Also keep in mind that the client or stakeholder

could be the culprit for the delays by going outside the normal channels of communication and inundating the designer with tweaks and last-minute changes.

The Developer

The developer is your secret weapon, your go-to player and most trusted resource on the project team. He or she thrives on mental challenges, brainteasers, and puzzles—the more difficult and convoluted the better. Beyond being technically proficient, exceptional developers can also see the forest for the trees. They quickly grasp the business goals behind the project (assuming the goals are valid) and can usually determine if the project will be a boon or a bust. And, given the opportunity, the developer can help *you* understand the project better as well.

When you embark on a project and you are creating the early drafts of the specifications, the developer will help you crystallize and define the back-end requirements. If you are not very technical, then save yourself a lot of frustration and potentially wasted time. Block out some time with the developer and collaborate on the technical requirements and functionality.

Watching a programmer design an application on a white board or even a scrap of paper can be a confusing but ultimately rewarding experience. With near religious zeal they will leave no stone unturned as they pepper you with questions and potential user scenarios you never thought of. They might embarrass you a little in the process by exposing the gaps in your knowledge or ideas, but the project will be so much the better for going through this exercise. After a white board session with the developer (or tech lead), you will begin to see your project from a 360-degree vantage point and will be able to go back to the producer or stakeholder and get the clarification necessary to answer questions, make suggestions, and meet their expectations.

The Tech Lead

The tech lead is your savior and friend. Cherish this person always and keep him or her close. Just as you are the bridge of communication between the technology staff and the stakeholder, the tech lead is the bridge between you and the high-voltage nether reaches of the developer's mind. Depending on the organization, the tech lead role can be a hands-on person, such as a senior developer, or a hands-off member of the technology department's managerial staff.

The tech lead is a great help during the technical design phase as you work out the backend specifications with the developer. The tech lead is also responsible for conducting code reviews and keeping developers on track during particularly thorny development projects. They are especially helpful when a junior or inexperienced developer is assigned to the project. The tech lead can also provide a buffer between yourself and a developer who is struggling and falling behind in the schedule or is not communicative.

The tech lead's deliverables and responsibilities can include the following.

- Technical specifications
- Code reviews
- Staff management
- Programming

If your company or department does not utilize a tech lead, it may be well worth your time to establish this role. Besides being a huge help for the project manager, the tech lead role can provide a career step for the development team. The role requires maturity and management skill and can provide a platform for acquiring both.

The Database Administrator

The database administrator (DBA) is one of the more specialized members of the team and is responsible for creating, advising, and controlling all aspects of the project that involve the database. The DBA is not typically a full-time member of the project team, but his or her contribution is invaluable to the project. The developer works closely with the DBA throughout the lifecycle of the build but more so at the outset of the project. Typically, a developer will create a database schema as one of the first steps in the technical design. The developer will then present the DBA with the schema, and the DBA will analyze it to be sure it meets the standards he enforces on the database. If the schema is not up to snuff, the DBA will consult with the developer on a more suitable schema for the project.

The DBA also writes code specific to the type of database being used for processes such as *stored procedures*. Once again, this work is done in collaboration with the developer assigned to the project. The bulk of the DBA's time is spent maintaining the database and optimizing its performance. The DBA's typical project deliverables and responsibilities can include

- Schema implementation
- Stored procedures and other database coding
- Staff management

The Quality Assurance Engineer

The QA tester is the final gateway between your project and life on the Web. Be kind to the QA department because the day will come when you will be negotiating with them for the release of your project and every advantage helps. Depending on the organization, QA testers can be very technical and troubleshoot bugs, or they may only test for poor user experience, design imperfections, and copy errors. Usually the QA tester is familiar enough with the technology to write up a coherent bug description but does not have the time or responsibility to get under the hood and investigate the cause of the bug. The QA tester should be involved early in the project and should be invited to all kickoff meetings. The QA department, like HTML, can be a bottleneck in the build process. Be

sure to get your QA tester involved early in the process to ensure that your project is in his job queue and on his radar. Even though the bulk of their tasks begin during the second half of the project (depending on the development approach), getting QA involved early gives them a preview of the moving parts and an idea of where potential design and technical flaws may occur.

The QA tester's deliverables and responsibilities can include

- Bug reports
- Creation and maintenance of a QA methodology
- Creation or procurement of a bug reporting tool

Summary

To successfully manage a Web development team, the project manager should be familiar with each team member's role and contribution. Although not an expert, the project manager is a generalist but must possess enough knowledge to be conversant with each resource. The following list of specialists is representative of a "standard" Web team.

- Project stakeholder/client
- Project manager
- Producer and/or content developer
- Information architect
- Graphic designer
- HTML developer
- Developer
- Tech lead
- Quality assurance engineer
- Project manager

To successfully manage your team, you must become a part of it both in deed and in spirit. Show your commitment to the team whenever you get the opportunity. Allow your people the freedom to arrive at their own solutions in their own way; don't micromanage or force a solution you might favor over one of their own devising.

See the next page for sample templates

Project Team²

Sequence Number	Initials	Responsibility	Start Date	Expected End Date
1	BB	Project manager	4 April 2000	3 November 2000
2	KP	Onsite coordinator	4 April 2000	3 November 2000
3	BJ	Module leader, backup project lead	15 May 2000	3 November 2000
4	SP	Configuration controller	22 May 2000	13 October 2000
5	DD	Developer	22 May 2000	29 September 2000
6	HP	Developer, backup configuration controller	22 May 2000	29 September 2000
7	NA	Developer	17 July 2000	3 November 2000
8	SH	Developer	1 August 2000	15 September 2000
9	SB	SQA	15 May 2000	3 November 2000

Roles and Responsibilities

Role	Responsibilities
Business manager (BM)	<ul style="list-style-type: none">• Resolve escalated issues• Review project status• Participate in critical technical reviews
Customer	<ul style="list-style-type: none">• Review design• Resolve escalated issues• Acceptance test planning and testing
Account manager (AM)	<ul style="list-style-type: none">• Customer satisfaction

² Software Project Management in Jalote, Addison Wesley, 2002

Role	Responsibilities
	<ul style="list-style-type: none"> • Business growth • Project financial plan • Interface with sales and marketing • Training-related issues • Employee-related issues
Project manager (PM)	<ul style="list-style-type: none"> • Project planning and scheduling • Design • Customer interaction • Reviews • Testing • Reporting • Task assignment and tracking • Ensure delivery as per contract • Interface with other departments as per need • Ensure open issues/customer complaints are closed properly • Ensure project members are adequately trained
Module leader (ML)	<ul style="list-style-type: none"> • Design • Development • Testing • Reporting
Developer (DV)	<ul style="list-style-type: none"> • Detail design for use cases • Development • Unit testing and integration testing
Software quality adviser (SQA) from the SEPG	<ul style="list-style-type: none"> • Process consultancy • Quality assurance (audits) • Install measurement tools and train project personnel • Participate in reviews of project plan and processes as necessary
Onsite coordinator	<ul style="list-style-type: none"> • Resolve any issues from customer/offshore • Support during development

Appendix A: Technical Specifications

You are required to submit Technical Specification for your Web Project. See an example of a mockup and the technical specifications in this appendix.

Wireframe Mockup (Calendar Applet)

Today is, Wednesday, February 23, 2000

Previous and Next links allow user to navigate within their accessible date range.

User range:
(Registration date=current date)

Previous Month

Next Month

To log in an entry for today, which is already selected just click done. To edit or input new data for past entry, click on the respective date.

S	M	T	W	T	F	S
			1 ...	2 ...	3 ...	4 ...

This takes user to the chart result page and displays user's past seven entries.

This calendar applet has color-coded days to alert user to the data entry status for each date.

The present date is selected by default. User can click on "done" to input data for the default date.

User can click on any date to edit or input new data for past entry.

All future dates are grayed out and not accessible to user.

This takes user to a list of the user's past seven journal entries.

User selects from the following options: weight, activity and food intake. This takes user to the graph result page.

Contains Data No Data Default Selectio Done

View your progress:

Chart

Graph Pull-Down Menu

When a reminder is set, it states the setting and frequency and allows user to edit or delete reminder.

First-time user will have this Reminder link included on all input and result pages.

User can go back and edit their profile.

Send to a friend



SeaState Internet Solutions

www.seastatesolutions.com

Technical Specifications Trend InfoWorld Web Site

NOTE: Wireframe mockups and screenshots included in this document are functional diagrams only, and are not intended to represent a proposed graphic design, page layout, or look and feel.

SECTION I: EXECUTIVE SUMMARY

- One-time Design and Development Costs: \$00,000.00
- Recurring Fees: \$000/month Hosting + \$000/month mailing list hosting
- Project Start Date: September 2000
- Project End Date: December 2000
- Project Duration: X weeks

Technical Objectives:

- Dynamically publish trend reports in article format online for the use of client subscribers.
- Provide a content management system that will allow editors to update the site's content via a Web interface.
- Provide a searchable archive of trend reports for the use of client subscribers.
- Allow administrators to manage a client subscription accounts database through a Web interface.
- Deliver the Deluxe Report in PDF format to subscribing clients.
- Allow clients to submit subscription requests online.
- Allow clients to view their subscription account details on the web.
- Provide a newsletter subscription sign-up and delivery system.
- Display general company and contact information.

Site Overview:

The site will contain the following major areas:

- Deluxe Report
- TrendInfoWorld Web Site
- General Company Information
- Content Management System
- Client Administration System

Hosting:

Hosting Fees:	\$000/ month
Time:	X weeks to configure Web server

- The hosting environment will be configured by SeaState Internet Solutions System Administrators, on a standard Microsoft Windows 2000 server running IIS 5.0 and a SQL Server 2000 database.
- Hardware specifications will be detailed and provided to the Hosting Provider Data Center.
- Operating system, Web server, e-mail and database software will be loaded onto the servers by the system administrators according to specifications.
- Once the servers are set up according to spec they will be tested. Once testing is complete the System will be migrated from the production servers to the hosted environment.

Domain Name Registration will point both domains to the same server.

<i>Computer</i>	Micron NetFrame 4400
CPU	600 MHz Pentium III
RAM	256 MB
Hard Drive	9 GB SCSI
IP Addresses	5
Monthly Bandwidth	50 GB

(etc. ...)

SECTION II : PAGE TEMPLATES

TrendInfoWorld Subcategory Page:

[See Wireframe Mockup](#)

Template Development Costs:	\$0,000
Development Time:	X weeks

This is the page that TrendInfoWorld subscribers will encounter after clicking on one of the subcategories in the navigational links. The purpose of this page is to promote articles within the subcategories and display content about the Subcategory. The page displays the following items:

- ❑ Subcategory Menu Navigation: Subcategories are dynamically pulled from the database and displayed in a DHTML menu box.
- ❑ Navigation links to 10 to 12 categories (music, fashion, sports, tech).
- ❑ Navigational links to the other sibling subcategories within this category.
- ❑ “You have x days remaining in your subscription” re-subscribe link .
- ❑ Browse the category by date and title
- ❑ Keyword Search (select category–“All” or any specific category).
- ❑ Company Info navigation.

HTML Prototype of DHTML Category Navigation:



TrendInfoWorld Search Results Page:

[See Wireframe Mockup](#)

Browse Category Archive by Title and Date:

Template Development Costs:	\$0,000
Search Engine Development:	\$0,000
Development Time:	X weeks

Full-Text Keyword Search Engine:

Template Development Costs:	\$0,000
Search Engine Development:	\$0,000
Development Time:	X weeks

The purpose of this page is to display TrendInfoWorld Search Results.

In the case of a keyword search, the search results will be displayed, 10 at a time, with navigational links to previous and next results. Results will be displayed with the article title hyperlinked to the full article, the category name, and the synopsis.

In the case of an Archive Browse by Title or Date, the Search Results Page will display a list of all the articles within a given category, 10 at a time, ordered by either Title or Date depending on the selection. Results will be displayed with the article title hyperlinked to the full article, the category name, and the synopsis.

Send This Page to a Colleague:

Simple E-mail Notification

Development Costs:	\$0,000
Development Time:	X weeks

Save E-mails to Database Add-on

Development Costs:	\$0,000
Development Time:	X weeks

This feature will be installed on all non subscription (free) pages. When the user clicks on the link, a JavaScript pop-up window appears that requests the following fields:

- Your Name
- Your E-mail
- Your Colleagues' E-mail
- Comments About This Site

On submit, an e-mail is sent to the recipient with a link to the referring page on the site. The message format is as follows:

```
SUBJECT: Check Out This Site
BODY:

[Sender's name] has recommended the following page from .com:

http://www..com/referrer...

Here are [Sender's name]'s comments about this page:

[show comments]
```

Additionally, the referred person's e-mail may be saved to the database and viewed in the Account Management System.

(etc. ...)

SECTION III : Content Management System

Development Costs:	\$00,000
Development Time:	4 weeks

The Content Management System is a password-protected area that allows company editorial staff to dynamically update content on the site.

Content Management System - Administrative Login Page:

This page presents a login screen that requests that the administrative user log in to the system. The login is stored in the database and may be changed. If the incorrect username/password is entered, then an error message will result.

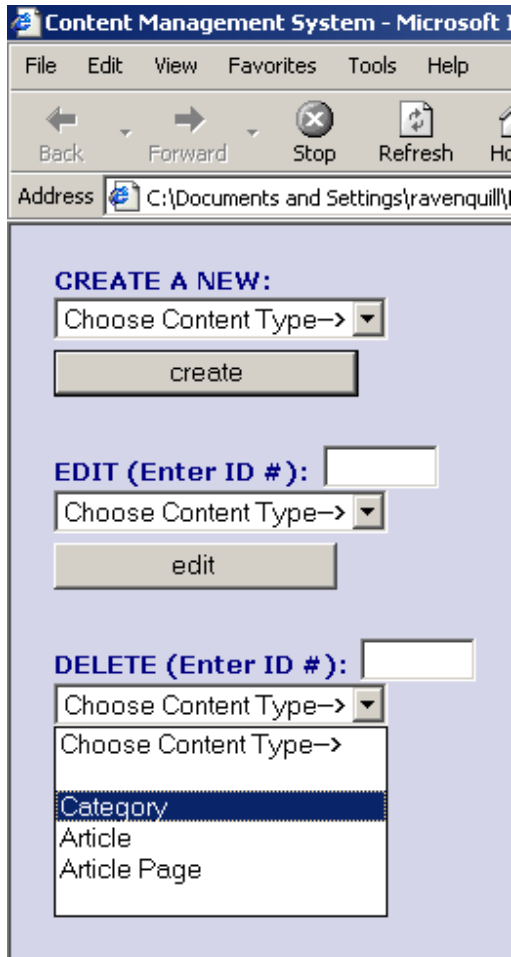
The main user actions or functions that administrators may execute are as follows:

- Create
- Edit
- List
- Delete
- Upload Image

Content Management System - Content Entry Functions:

- *Create*: By selecting this option the administrative user may create a new content item. The system will assign an ID automatically.
- *Edit*: This action allows the administrative user to modify a currently existing content item. This action requires that the administrative user know the content ID, which can be obtained from the List feature. The administrative user may edit any content item.
- *List*: This function allows user to view all of the content of a certain type as well as its IDs.
- *Upload Image*: This option allows the administrative user to upload an image directly to the Web server, using the ASP upload component. The image is uploaded from the user's hard drive directly to the common directory `/images/Articles/`. The confirmation page displays the image source HTML tag that can be used to insert the image onto article pages.
- *Upload Latest Deluxe Report*: This feature allow the user to upload a new Deluxe Report PDF file to the Web server, using the ASP upload component. This form also contains a "Date Published" Field.
- *Upload Sample Deluxe Report*: This feature allow the user to upload a new Deluxe Report PDF file to the Web server, using the ASP upload component. The user will be prompted to overwrite the current filename so that the sample report link will automatically point to the new file.

- *HTML Guide*: This is a link to basic HTML instructions that will be useful in content entry.

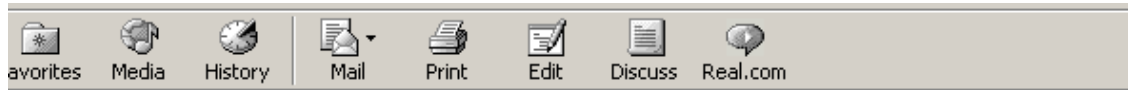


Content Management System - Content Items:

The content items in the database that the user may act upon using the functions listed above include:

- Home page copy
- Category
- Subcategory
- Article

(etc...)



nes.asp

Please assign the Article to one or more Categories

- Charts
- City Guides
- Entertainment
- Lifestyle

Article Date: (50 chars. max)

10/10/2001 7:12:34

Don't Show the Article Date

Article Keywords: (500 chars. max)

keyword1, keyword2, keyword3

SECTION IV : PRELIMINARY DATABASE SCHEMA

Content Management System Database Schema:

Trend Subcategory Page Schema: The Trend Subcategory pages will contain the following fields:

- ❑ ID
- ❑ Subcategory Name
- ❑ Keywords (also populates meta tag)
- ❑ Synopsis (also populates meta description)
- ❑ Body

```
TblTrendSubCategory
ID
TrendCategoryID (Foreign Key to tblTrendCategory)
TrendSubCategoryName (50)
TrendSubCategoryKeywords (600)
TrendSubCategorySynopsis (600)
TrendSubCategoryBody (varchar 10000)
```

(etc. ...)

SECTION V: Phase II Enhancements

This section details future enhancements that are outside the scope of this launch:

Newsletter Enhancement Option 2: HTML Newsletter:

Development Costs – HTML Newsletter Delivery:	\$00,000
Development Time – HTML Newsletter Delivery:	X weeks

HTML newsletters will require a more costly back-end, comprised of the following components:

- Newsletter creation user interface: This system will be part of the Content Management System. The user will enter content into fill-in form fields which will be saved to the database. This content will then be inserted into an HTML page template.
- Newsletter HTML template
- Newsletter preview: The interface will have a preview feature that will allow the user to preview the HTML page before sending it out.
- Newsletter send command: This will allow the user to send out a newsletter once it has been created and previewed. The user will specify the Newsletter ID and click “Send”
- Newsletter archive: The database will save all newsletters that have been created on the system. These archived newsletters can be viewed via the “List” function.
- Newsletter mailing engine: A Perl script will set the MIME type headers to indicate that the newsletter is an HTML newsletter, and then send the processed HTML page to the mailing list.

Preview HTML Newsletter: This feature allows the user to preview an HTML newsletter before sending it out to subscribers. This feature takes the newsletter ID as an argument and invokes the Newsletter Drawing Template, displaying the finished page.

Send HTML Newsletter: This feature allows the user to designate the ID of the HTML newsletter that they would like to send out. By clicking “Send,” the database populates the HTML newsletter template and invokes the perl mailer script.

HTML Newsletter: These fields are published to the HTML newsletter template:

```
tblNewsletter
ID
NewsTitle (varchar 100)
NewsBody (varchar 10000)
```

(etc.)

Trend InfoWorld Subcategory Page

Subcategories appear (DHTML)

Categories:

Beauty

Style

Entertainment

- News
- Celebs
- TV
- Film

Sports

Music

Gadgets

Internet

Lifestyle

Cities

Youth Events

Stats & Facts

SEARCH

You have x days left in your subscription, [Re-subscribe Now](#) |

Browse

Archive

Browse the subcategory by :

- Title
- Date

Subcategory Name

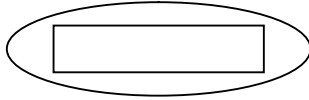
TrendInfoWorld Subcategory Body Text

This is a block of HTML and plain text that can introduce the Subcategory (e.g., "TV"), promote articles within the Subcategory, or link to free standing articles or external sites.

[Send This Page to a Colleague](#)

[About Us](#) | [Products](#) | [Services](#) | [Clients](#) | [Press](#) | [Contact Us](#)

TrendInfoWorld Search Results Page



You have x days left in your subscription,
[Re-subscribe Now](#) | [My Account](#)

Trends Central SEARCH

More Results: [Prev](#) | [1](#) | [2](#) | [3](#) |
[Next](#)

Browse
Archive

Browse by :
▪ Title
▪ Date

SEARCH

Enter Keywords

Select Category Name

Trends Article Title: *Category Name, Month Day Year*
Synopsis Text. This is the synopsis text pulled from the database. This text should be about 75 words long.

Trends Article Title: *Category Name, Month Day Year*
Synopsis Text. This is the synopsis text pulled from the database. This text should be about 75 words long.

Trends Article Title: *Category Name, Month Day Year*
Synopsis Text. This is the synopsis text pulled from the database. This text should be about 75 words long.

Trends Article Title: *Category Name, Month Day Year*
Synopsis Text. This is the synopsis text pulled from the database. This text should be about 75 words long.

More Results: [Previous](#) | [1](#) | [2](#) | [3](#) | [Next](#)

[About Us](#) | [Products](#) | [Services](#) | [Clients](#) | [Press](#) | [Contact Us](#)

Appendix B: Software Project Management Documentation Examples³

Please browse the following examples. You may want to consider them as deliverables for your Web Project.

³ Software Project Management, Jalote, Addison Wesley Professional 2002.

The Change Management Process

During project planning, a project manager decides which process is to be followed for handling change requests. The planned process is discussed with the customer so that both the customer and the vendor are in agreement about how to manage changes. Generally, the process specifies how the change requests will be made, when formal approvals are needed, and so on. When a request for a requirements change comes in, the requirements change management process must be executed.

Because change requests have cost implications, it is necessary to have a clear agreement on payment. Frequently, with customer approval, projects build a buffer into their estimates for implementing change requests (typically a small percentage of the total project effort). Such a budget provision simplifies the administrative aspects of implementing approved change requests. Sample steps you must perform:

1. Log the changes.
2. Perform an impact analysis on the work products.
3. Estimate the effort needed for the change requests.
4. Re-estimate the delivery schedule.
5. Perform a cumulative cost impact analysis.
6. Review the impact with senior management if thresholds are exceeded.
7. Obtain customer sign-off.
8. Rework work products.

You maintain a change request log to keep track of the change requests. Each entry in the log contains a change request number, a brief description of the change, the effect of the change, the status of the change request, and key dates. You assess the effect of a change request by performing impact analysis. Impact analysis involves identifying work products that need to be changed and evaluating the quantum of change to each; reassessing the project's risks by revisiting the risk management plan; and evaluating the overall implications of the changes for the effort and schedule estimates. The outcome of the analysis is reviewed and approved by the customer.

A change might be classified as minor if the total effort involved in implementing it does not exceed a predetermined value—say, two person-days. Minor changes typically become part of the project effort, utilizing the buffer in the planned estimate. Major changes usually have a larger impact on effort and schedule and must be formally approved by the client.

Quality Planning

Review Point	Review Item	Type of Review
End of project planning	Project plan Defect control system set up Project schedule	Group review Software quality adviser review Software quality adviser review
End of project planning	CM plan	Group review
End of 90% of requirements (this should be at the end of the first elaboration iteration)	Business analysis and requirements specification document Use case catalog	Group review
End of 90% design (this should be at the end of the second elaboration iteration)	Design document Object model	Group review
Beginning of each iteration	Iteration plans	One-person review
End of detailed design	Complex and first-time-generated program specs including test cases, interaction diagrams	Group review
After coding of first few programs	Code	Group review
After self-testing of a process	Code	One-person review
End of unit test plan	Unit test plan	One-person review
Beginning of integration test	Integration test plan	Group review

Milestone Commitments

Sequence Number	Milestone Date	Milestones	Deliverables
1	26 May 2000	Inception: Requirements sign-off	Business analysis and requirements specifications, use case catalog, screens, iteration plan
2	15 May–23 June 2000	Elaboration: Iteration 1	Sequence diagrams, class diagram, source code, plan for the next cycle
3	26 June–7 July 2000	Elaboration: Iteration 2	Supplementary specifications, sequence diagrams, class diagram, architecture document, source code, iteration plan for the next cycle
4	10 July–21 July 2000	Construction: Iteration 1	Source code, review reports, test reports, iteration plan for the next cycle
5	31 July–8 Aug 2000	Construction: Iteration 3	Source code, review reports, test reports, iteration plan for the next cycle, deployment plan for the product
6	9 Aug–1 Sep 2000	Integration testing phase	Test plans, test reports
7	4 Sep–15 Sep 2000	Code delivery and setup	Code
8	2 Oct–26 Oct 2000	Acceptance test	Test results
9	27 Oct–3 Nov 2000	Rollout and support	Project sign-off

Other Commitments

Sequence Number	Commitments
1	This project will follow the Rational Unified Methodology (RUP).

Assumptions Made while Planning

- Migration to Visual Age for Java 3.0 will not be done by this team.
- Qualified people will approve Rational Unified Process methodology for implementing this project.

Technical Architecture

Development Environment

Hardware	Software
NT Server	Win NT
MainFrame	DB2
Intel PC	VisualAge for Java, Java, Win NT

Hardware and Software Resources Required

Item Description	Required #	Date
PCs with 128 RAM	6	1 May 2000
1GB space on server	1	1 May 2000
VisualAge for Java	6	4 May 2000
DB2	6	4 May 2000
Rational Rose	5	15 May 2000
Requisite Pro	1	15 May 2000

Tools

Tools List	
Tools to be developed in the project	None

Training Plan

Training Area	Duration	Waiver Criteria
Technical		
Java Language	7 days	If already trained
Dreamweaver	3 days	Exposed as part of initial training

Strategy for Meeting Quality Goals

Strategy	Expected Benefits
Do defect prevention using the standard defect prevention guidelines and process; use standards developed in Synergy for coding.	10–20% reduction in defect injection rate and about 2% improvement in productivity
Group review of program specs for first few/logically complex use cases. Group review of design docs/first time-generated code by project leader, developer, and one consultant.	Improvement in quality as overall defect removal efficiency will improve; some benefits in productivity as defects will be detected early
Introduction of RUP methodology and implementing the project in iterations. Milestone analysis and defect prevention exercise will be done after each iteration.	Approximately 5% reduction in defect injection rate and 1% improvement in overall productivity

Reviews

Review Point	Review Item	Type of Review
End of project planning	Project plan DCS set up Project schedule	Group review SQA review SQA review
End of project planning	CM Plan	Group review
End of 90% of requirements (This should be at the end of first elaboration iteration)	Business analysis and requirements specification document, Use Case catalog	Group review
End of 90% design (This should be at the end of second elaboration iteration)	Design document, object model	Group review
Beginning of each iteration	Iteration plans	One-person

Review Point	Review Item	Type of Review
		review
End of detailed design	Complex/first time generated program specs incl. test cases, interactive diagrams	Group review
After coding for first few programs	Code	Group review
After self-testing of a process	Code	One-person review
End of unit test plan	Unit test plan	One-person review
Beginning of integration test	Integration test plan	Group review

Quality Tracking

Quality Activity	Action
Defect tracking	Use DCS for logging defects and tracking them to closure.
Reviews (requirements, high-level design, detailed design)	Check against project goals in quality plan.
Code review	Check against limits for each program through SPC tool.
Independent unit testing	Check against limits for each program through SPC tool.
Integration testing/System testing	Check against project goals in quality plan.

Status Reporting

Review by Senior Management (BM)

Sequence Number	Item for Review	Frequency of Review
1	Schedule	Every version change
2	Project plan	When significant changes are made
3	Milestone report	End of milestones

Status Reporting

Report To	Frequency
Business manager	Weekly on Monday by e-mail
Customer	Weekly on Monday

Deviation Limits at Milestones

Actual vs. Estimated of:	For the First Five Milestones	For the Rest of the Milestones
Effort	10%	5%
Schedule	10%	5%
Defects	20%	20%

Appendix C: Closure Report⁴

Finally you are required to submit a closure report.

Please consider the examples you see in this appendix for your closure report.

⁴ Software Project Management in By Jalote
Addison Wesley Professional 2002

Closure Report

A project does not end with the delivery and installation of the software; before it is closed, it must be used for learning. Project closure analysis is one method to achieve this goal. Following are some of the key takeaways from the Infosys approach to project closure:

- Keep the project closure analysis metrics-based. Analyze the data to understand the performance of the project and the causes for any major deviations. These causes can serve as a source of improvement initiatives.
- The metrics analysis should report the final quality delivered, the productivity achieved, the distribution of effort, the distribution of defects, the defect removal efficiency of various quality activities, and the cost of quality.
- Collect reusable process assets such as plans, checklists, standards, and guidelines, and make them available for others.

GENERAL INFORMATION

Project Code	Xxxxx
Life Cycle	Development, Full life cycle
Business Domain	Finance. Web-based application for managing accounts.
Project leader/Module Leader	Xxxxxx
Software Quality Adviser	Xxxxx

2. PERFORMANCE SUMMARY

Performance Parameter	Actual	Estimated	Deviation	Reasons for Deviation (If Large)
Total Effort (person-days)	597	501	19%	Two major change requests that came.
Peak Team Size	9	9	0	N/A
Start Date	03 Apr 2000	03 Apr 2000	0	N/A
End Date	03 Nov 2000	30 Nov 2000	27 Days	Two major change requests consumed more than 5% of the effort.

Performance Parameter	Actual	Estimated	Deviation	Reasons for Deviation (If Large)
Quality (number of defects delivered per FP)	0.002	0.0125		Quality improved because of defect prevention and use of incremental process.
Productivity	58	57	2%	N/A
Cost of quality	31.4%	33%	5%	N/A
Defect injection rate	0.022	0.03	-26%	Improved because of defect prevention.
Defect removal efficiency	97.4	97	Small	N/A

3. PROCESS DETAILS

Process Tailoring	<ul style="list-style-type: none"> • Rational Unified Process was employed. • Development and analysis were done iteratively—3 iterations for development and 2 for design and analysis. • Requirement traceability was done through Requisite Pro tool.
-------------------	---

4. TOOLS USED

Notes on Tools Used	<ul style="list-style-type: none"> • External Tools: VSS, VJA, Requisite Pro, MSP • Internal Tools: BugsBunny, WAR
---------------------	--

5. RISK MANAGEMENT

Risks identified at the start of the project

Risk 1	Lack of support from database architect and database administrator of the customer
Risk 2	Improper use of RUP, as it is being used for the first time
Risk 3	Personnel attrition
Risk 4	Problems with working on customer's database over the link

Risks encountered during the project

Risk 1	Impact of conversion to VAJ 3.0
Risk 2	Lack of support from database architect and database administrator of the customer
Risk 3	Improper use of RUP, as it is being used for the first time
Risk 4	Personnel attrition

Notes on Risk Mitigation

Risk1: Clearly articulating the risk helped in customer agreeing to postpone the conversion with proper budgeting of its impact.

Risk2: Mitigation strategies of careful and advance planning and employing the on-site coordinator were effective.

Risk3: Training the team in RUP was effective. So was keeping the customer informed.

Risk 4: Remained as a risk, although it did not materialize. Impact would have been minimal because multiple people were kept informed of each critical activity.

6. SIZE

	Estimated	Actual
Number of simple use cases	5	5
Number of medium use cases	9	9
Number of complex use cases	12	12

Notes on Estimation

Classification Criteria. The standard definition of simple, medium, and complex was used for classifying the use cases. This worked fine.

Final System Size in FP

The size of the final source is measured in LOC. It is normalized to FP by using the published conversion tables. For Java, the published tables suggest that 21 LOC equals 1 FP and for COBOL, 107 LOC equal 1 FP.

Output Language	Size in LOC	Size in FP
Java	33,865	1612
COBOL	1241	12

7. SCHEDULE

Phase	Actual Elapsed Time (days)	Estimated Time (days)	% Slippage	Reasons for Slippage
Requirements	28.67	31	-6.5	
High-level design	0	0	0.0	
Detailed design	38.8	42	-6.7	
Coding	132	135	-1.6	
Unit testing	9	10	-9.3	
Total - Build	141	144	-2.1	
Integration test	40	40	0	
System testing	15	0	0.0	
Acceptance testing	30	10	200.0	AT completion was extended on customer's request.

8. EFFORT

Distribution over Life-Cycle Stages

Stage	Task	Review	Rework	Total
Requirements	210.0	10.0	60.0	280.0
High-level design	0.0	0.0	0.0	0.0
Detailed design	652.0	14.0	29.5	695.5
Coding	1188.0	39.5	76.5	1304.0
Unit testing	129.5	0.0	17.0	146.5

Stage	Task	Review	Rework	Total
Integration testing	567.5	6.0	160.5	734.0
System testing	90.0	0.0	0.0	90.0
Acceptance testing	336.5	0.0	0.0	336.5
Total - LC stages	3173.5	69.5	343.5	3586.5
Project management	733.1	0.0	0.0	733.1
Training	104.5	0.0	0.0	104.5
CM	317.0	0.0	0.0	317.0
Misc.	488.5	0.0	0.0	488.5
Total – mgmt, training, and misc.	1643.0	0.0	0.0	1643.0
Total Effort (Person-hours)	4816.50	69.50	343.50	5229.50
Total Effort (Person-months)	25.76	0.37	1.84	27.97

Cost of Quality

$$\begin{aligned}
 \text{COQ} &= \frac{\text{Review effort} + \text{rework effort} + \text{test effort} + \text{training effort}}{\text{total effort}} = 100 \\
 &= (69.5 + 343.5 + 129.5 + 567.5 + 90 + 336.5 + 104.5) / 5229.5 \times 100 \\
 &= 31.4\%
 \end{aligned}$$

Effort Distribution and Actual Versus Estimated

Stage	Actual		Estimated		% Deviation	Reasons or Deviation
	Effort (person-hours)	%	Effort (person-hours)	%		
Requirements	280	5.35	475.0	10	-30	Overestimated this effort (data from earlier project did not help because it did not have this phase).
Design (HLD)	695.5	13.30	569.0	12	22	Design took more time

Stage	Actual		Estimated		% Deviation	Reasons or Deviation
	Effort (person-hours)	%	Effort (person-hours)	%		
and detailed)						because team was inexperienced with Rational Rose and OOAD.
Coding	1304.0	24.94	1235.3	26	6	
Unit testing	146.5	2.80	142.5	3	3	
Integration testing	734.0	14.04	331.0	7	120	Much effort spent on fixing bugs introduced during reconciliation with Synergy and Window Resized code.
System testing	90.0	1.72	95.0	2	-5	
Acceptance testing	336.5	6.43	285.0	6	18	Acceptance testing was not completed on Nov 3 and was extended until Nov 23 due to delays from the customer.
Total—LC stages	3586.5	68.58	3132.8	66	14.5	
Project management	733.1	14.02	713.0	15	3	
Training	104.5	2.00	455.0	10	-77	
CM	317.0	6.06	142.0	3	123	Deviation due to reconciliation issues.
Misc.	488.5	9.34	285.0	6	71	More because of training.
Total—mgmt, training, and misc.	1643.0	31.42	1595.0	34	3.01	
Total	5229.5	100	4727.8	100	10.6	

9. CAUSAL ANALYSIS AND LESSONS LEARNED

There were very few large deviations in the process performance; the actual performance was close to what was expected. The reasons for the deviations, where they are large, are given along with the deviation. Some key lessons learned are:

1. Incremental or phased development is extremely helpful in achieving higher quality and productivity because data from the first phase can be used to improve the remaining phases through defect prevention.
2. Defect prevention can substantially reduce the defect injection rate. In terms of effort also, defect prevention pays off handsomely; by putting in a few hours of effort, up to 5 to 10 times effort savings can be achieved in the form of reduced rework effort.
3. If a change request has a major impact, discussion with the customer using a detailed impact analysis can be very helpful in setting the right expectations and doing a proper cost-benefit analysis (which may result in postponement of the change, as happened in this project).
4. The defect removal efficiencies of code reviews and unit testing are very low. Processes for both, and implementation of these processes, need to be reviewed to improve these numbers. In this project, the system/integration testing compensated for the poor performance of reviews and unit testing. However, for larger projects, this may not be possible and poor performance in reviews and unit testing can have adverse effects on quality.

11. ASSETS SUBMITTED

Project management plan, project schedule, configuration management plan, Java coding standards, code review checklist, integration plan review checklist, impact analysis checklist, causal analysis reports for defect prevention.

Other Resources

Please visit <http://office.microsoft.com/en-us/assistance/CH010397741033.aspx> it gives you additional information on creating a Project Plan.

