Governments in every country have realized that there are many benefits of using IT systems to provide government services to its citizens. Some of the main benefits of using IT systems in delivering government services include:

- **Transparency**: Policies implemented at central government level has to be implemented at grass roots level using multiple levels of governance. This results in lack of transparency as reports of execution at grass roots level is difficult to be reported at each level of governance and finally to be reported at the central government. IT systems make it possible to show results and reports at each level of government easily and transparently.

- **Corruption avoidance**: any government system can be manipulated and corruption can seep in if any government policy is not implemented transparently. IT systems bring transparency in executing government policies and thus help in avoiding corruption.

- **Recipient identification**: It is very difficult to identify recipients of government policies. This results in giving government services to wrong set of citizens who do not deserve the benefits. Once a database of right beneficiaries is identified and made in an IT system then chances of wrong recipients receiving government benefits is minimized.

- **Policy coverage**: very often government policies are not able to cover all segments of recipients because of lack of data available to the government agency. Once recipient data is available with the government in their system databases then 100% policy coverage can be achieved.

- **Speedy policy delivery**: each time a new government scheme is launched, a database of right recipients need to be created. This results in very slow delivery of government policy as database creation takes a lot of time. If the database is already available in government IT system then the delivery process will be fast and smooth.

- **Policy making**: governments make new public policies based on collected information about citizens. The process of receiving such information is very slow. If information about citizens already exists in an IT system then governments can make new policies faster.

Realizing these benefits of IT systems, governments are keen on implementing IT systems in every government departments and government levels. This has resulted in increased government spending on IT & software systems.

In India, three major decisions taken by the central government about implementing IT systems include

- **UIDAI (Unique Identification Development Authority of India)**: the central government created a new government department and named it UIDAI. This is an ambitious initiative to identify each and every citizen of India by taking personal identification data and making a central database. This initiative will help the central government as well as local government administration agencies to identify right recipients for all government spending on health, food, education etc.
Police record management system: Under this scheme, all police stations will be required to keep First Information reports (FIRs), crime statistics etc. in the central IT system. This system will allow citizens to make complaints at local police stations even without visiting the concerned police station by registering FIRs online. Many police stations refuse to register FIRs by citizens on flimsy grounds and this facility will eliminate this malpractice.

Case management & court administration system: The long running court cases generate a lot of paper documents. Managing them is messy. All courts at district, state and supreme court of India will be provided with a case management & court administration system so that all problems associated with paper managed systems will be removed.

Projects are very different from other kinds of processes. For example, a manufacturing process is almost continuous. A manufacturing plant keeps manufacturing the goods. The resources like manpower, machinery, raw materials etc. for any manufacturing plant is almost constant when measured against time.

Projects on the other hand have a definite start date and end date. At various points in time, a project may require different amount of resources. For example on a software project, a large number of software engineers may be needed when the project is undergoing construction phase. But at other phases of the project, it may need less number of resources.

Projects also need team members with different skill sets. For example on a software project, software requirement specification specialists, software designers, software engineers (developers), software testers may be needed.

One more aspect about projects is that projects go through various phases where different kinds of work may be performed. For example on a software project, software modeling work will be performed when the project phase is going through a software design phase. This work will not be performed in other phases of the project.

Chapter 2

Misunderstanding between the project stakeholders & the project team is always possible. Project team comprises of people who work on designing, building and testing software products. On the other hand, project stakeholders are people from business side. They are exposed only to business and have not much exposure to software. Due to differences on outlook and perspectives, there always exists a possibility of misunderstanding between the project team and the project stakeholders.

To mitigate the risk of misunderstanding with project stakeholders, the project team must take initiatives.
Some of the common points on which misunderstanding happens include:

- **Language**: project team members lack knowledge of business language that project stakeholders speak. So they do not understand what is being stated by project stakeholders. At the same time, what is stated by project team is not understood by project stakeholders.
- **Business knowledge**: project team lacks business knowledge and thus is not able to comprehend requirements of project stakeholders.

To overcome these difficulties, the project team members should take initiatives including the following:

- **Learn business language**: project team members must learn the business language spoken by project stakeholders. For example, project stakeholders may state that they want a scheduling system which will schedule their campaigns for making various steel grades. If the project team has never worked on any project involving steel making, they will never understand what a campaign is all about. So the project team must get conversant with the business language spoken by project stakeholders.
- **Acquire business knowledge**: business knowledge is very important so that appropriate software solution is created as per user requirements. For example, understanding campaigns in steel making will help project team members to create appropriate business solution.
- **Listening comprehension**: what is being stated by project stakeholders must be listened carefully by the project team. In many cases, project team is too busy in talking and misses on listening to what the project stakeholders are stating.

Due to these factors, it makes sense to have business analysts on the project team for all meetings with project stakeholders. Business analyst people have worked in the same industry in which the project stakeholders operate and thus understand the business language as well as have deep knowledge of the industry.

**Answer 2.2**

**Open source project:** [www.cyclos.org](http://www.cyclos.org)

Cyclos is a project of a network of Social Trade Organisations from Latin America and Europe.

**Project charter:**
The objective of the Social Trade Organisation is to provide a free (or low cost) platform that can be used by different organisations and institutions that are aiming on social economic impact.

Cyclos is the platform through which Social Trade organization will achieve its objectives.

**Open source project:** [httpd.apache.org](http://httpd.apache.org)

**Project charter:**
Software Project Management: A Process Driven Approach

The Apache HTTP Server Project is a collaborative software development effort aimed at creating a robust, commercial-grade, featureful, and freely-available source code implementation of an HTTP (Web) server.

Chapter 3

Answer 3.1
Effort required for a software project is derived from the productivity, size and complexity of the software product being built. Cost for the software project can be derived from the required effort and average salary of the software engineers deployed on the project.

There are also other costs on the project like office rent, hardware cost, management cost, travel cost, software tools cost etc. These costs are added to the effort cost to calculate total project cost.

Exact project cost and breakdown of project costs can be done using ABC cost analysis. Here each cost head (activity) is defined and whenever a cost occurs on the project then it is recorded against appropriate head.

Answer 3.2
Traditional project development requires a large amount of management overhead. It also requires a large amount of documentation. These types of projects also keep a reserve for unexpected change requests. These overheads result in excess cost required for these projects.

There are many management levels required on traditional projects. For example there will be a project leader for software testing team and another project leader for the development team. There could be some more management staff under the project manager, working on the project. This leads to a large management overhead.

Traditional projects require a large amount of documentation. It is because each and every project activity needs to be documented for later reference. Compare this with agile projects where they do not have any documentation at all.

Agile projects do not keep any reserve for unexpected changes in software design or requirements. It is because, software development happens in constant presence of the customer (customer on site). Each project iteration is completed and signed off by the customer. So there is no need to keep any reserve for unexpected changes. In contrast, traditional software projects are initially approved by the customer but after progress starts, there is no involvement of the customer. At later stages, the customer can realize that some changes in requirements are needed. To take care of such necessities, the project manager must keep some reserve in the project for requirement or design changes and consequently corresponding extra effort needed for software construction and testing.

Chapter 4
Answer 4.1
Whether the promised software product will be realized at all after spending so much time and money; is the great worry most software project sponsors deal on traditional projects. Project sponsors have so many stakes hanging on success of any software project. For most large businesses today, success of their IT strategy plays an important role. If any of their software project initiative fails then they along with their business faces insurmountable losses.

For this reason, they always want to know if the software project they initiated is on track or is failing somewhere. Traditional software development models do not provide any means through which the project sponsors can check if the software product being made is actually what they wanted. During the product development all they get is reports about how much progress has been done on the project but are never able to see any software product in reality. For example a project is planned to be completed in 6 months. Even at the end of 5 months, project sponsors may have no idea about how the software product looks like. It is because by this time the software construction is in progress and there is no way to see how the software product looks like. They will be able to see the software product only after 6 months when the product is fully constructed and tested. This indeed poses a great risk.

Iterative development models were introduced to address this very issue. Software is developed incrementally so that at every increment, the project sponsors can see the actual working software product. This assures them that their money is being spent wisely.

Due to nature of iterative projects, some different kinds of risk arise. Some of them include:
- No documentation: All communication on iterative projects is done through meetings and so communication is mostly verbal. When any changes need to be done after a product has been developed then it will be almost impossible to find out basis of already developed software. It is because no documents were maintained.
- Incoherent user requirements: users tend to refine their requirements over time. At first cut, their requirements are usually not coherent. On agile projects it is not possible to go back and refine requirements. This may lead to problems for the development team.
- Large projects: it is difficult to build large projects within short period of time as iterative project teams are very small.
- Product architecture: there is no elaborate product architecture designed for the entire product at the beginning of the project. When the size of product becomes large then there is no supporting infrastructure in form of completely balanced product architecture in place. This leads to shaky product which might not be able to support the big size due to lack of sound architecture to back it.

Answer 4.2
Many internal or external risks can not be managed even though a majority of other risks can be managed on projects. Some of the risks which can not be managed include:
- Customer going out of business
- Technology obsolescence
A risk can be managed only when some alternatives are available to the project manager. If no alternatives are available then the risk can not be managed. For example, if the customer is going out of business then there are no alternatives to replace the customer. The customer has been financing the project and the project work was being done for the customer. If customer is going out of business then there is no way the project can proceed further.

If some hurricane strikes the city and subsequently vast devastation occurs which leads to unrecoverable loss to life and kind then the project can not proceed further. Likewise if a vital tool was being used on the project and the tool becomes obsolete and is no longer being supported by the vendor and if no comparable tool is available which can replace it then it can lead to project not been able to proceed further.

In all such cases, the risk can not be managed and the project can not proceed further.

Chapter 5

Answer 5.1

description: this configuration system is for Index of chromium-browser-continuous/Win_x64/209752/ build. The main branch is for chrome-win32-sym.zip.
   file size is 173 MB. The change log is in XML format. A continuous build system was used for building this product. The revision history is also maintained.

description: this build is for closure compiler for chrome. The Closure Compiler is a tool for making JavaScript download which makes the download faster. It is a true compiler for JavaScript. Instead of compiling from a source language to machine code, it compiles from JavaScript to better JavaScript. It parses your JavaScript, analyzes it, removes dead code and rewrites and minimizes what's left. It also checks syntax, variable references, and types, and warns about common JavaScript pitfalls.

Answer 5.2

a. Last 20 finished builds for chrome as on 12 November 2014 (http://build.chromium.org/p/chromium/one_line_per_build):

<table>
<thead>
<tr>
<th>Time</th>
<th>Revision</th>
<th>Result</th>
<th>Builder</th>
<th>Build #</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Nov 12 10:20</td>
<td>3398861d8c627830eb72e238e69fleeaf570ddbe2</td>
<td>success</td>
<td>Mac</td>
<td>#30854</td>
<td>Build successful</td>
</tr>
<tr>
<td>2. Nov 12 10:44</td>
<td>0e54a112d3cecf9992d67ec0742ed3d591db691</td>
<td>success</td>
<td>Linux x64</td>
<td>#73390</td>
<td>Build successful</td>
</tr>
<tr>
<td>3. Nov 12 10:34</td>
<td>0e54a112d3cecf9992d67ec0742ed3d591db691</td>
<td>success</td>
<td>Android</td>
<td>#30924</td>
<td>Build successful</td>
</tr>
<tr>
<td>4. Nov 12 10:29</td>
<td>fc35d04e8e3c3eb520206fa6640a32714867ed7b2</td>
<td>success</td>
<td>Linux x64</td>
<td>#73398</td>
<td>Build successful</td>
</tr>
<tr>
<td>5. Nov 12 10:11</td>
<td>3398861d8c627830eb72e238e69fleeaf570ddbe2</td>
<td>success</td>
<td>Win</td>
<td>#26202</td>
<td>Build successful</td>
</tr>
<tr>
<td>6. Nov 12 10:04</td>
<td>e4016b8bb73b9ad385174e1c11cb394b4211e0</td>
<td>success</td>
<td>Linux</td>
<td>#55285</td>
<td>Build successful</td>
</tr>
<tr>
<td>7. Nov 12 10:09</td>
<td>3398861d8c627830eb72e238e69fleeaf570ddbe2</td>
<td>success</td>
<td>Android</td>
<td>#30923</td>
<td>Build successful</td>
</tr>
</tbody>
</table>
Chapter 6

Answer 6.1

Project planning in construction industry differs completely from that for software industry. In construction industry, procurement, supply, assembly, engineering, erection and commissioning are major activities carried out. The first job on these projects is to do a preliminary engineering design for buildings and machinery. Based on the preliminary design, major contractors are invited for making bids. The contractor who wins the bid then prepares the detailed designs. Procurement activities are then carried out for all required equipments and machinery. Afterwards assembly, erection, commissioning etc. activities are carried out. Trial is then done for all machinery which is installed. Once trial runs are successful then the contractor hands over machinery & buildings to the project sponsors. The sponsors close the project and contractor winds up its setup from the construction site.
Construction projects are resource intensive projects. They need specialized machinery, skilled manpower and huge quantities of construction material. Most of the budget on construction projects is for procuring machinery and materials. Cost of manpower on these projects come not more than 2-3% of the total project budget and majority of the budget is consumed in procuring material and machinery (more than 85 – 90%).

Software projects are different in many respects compared to construction projects. Software projects do not need much of machinery or material. They mostly need very skilled manpower. The project team is responsible for creating and delivering the software product at the end of the project as per specifications provided by the customer. The project team creates specifications for designing the software product, builds and tests the software product and at the end of the project, hands the software product to the customer. Majority of project budget is consumed in paying salaries to project team (more than 80% of project cost).

Answer 6.2

Types of tasks on software projects include building requirement specifications, software design and implementation and software testing. When project planning is done, each of the tasks associated with each type of tasks mentioned above are broken down to the smallest possible pieces of tasks. For example the construction of the product is broken down to the smallest features of the product. These pieces are then assigned to individual software engineers. Suppose a software engineer is lagging behind the schedule and the project manager is forced to think of engaging one more software engineer and assigning him/her to the new engineer. In most cases, it is difficult to do so. It is because the pieces of tasks assigned to the first engineer are scheduled one after the other. The pieces of tasks scheduled to be done later in the project can not be taken earlier because they depend on completion of the tasks scheduled earlier. Once the earlier tasks lag behind schedule then it is impossible to complete the later tasks also ahead of schedule.

Most tasks on software projects which could be done in parallel have already been planned and resources assigned. Similarly, tasks which can never be done in parallel and need to be done one after another have also been planned and resources have been assigned.

Now it is impossible to change the way tasks have been assigned on the project. That is why tasks on software projects are considered inelastic and at the same time duration for all of these tasks can neither be stretched nor shrunk by adding or removing people on the project.

Chapter 7

Answer 7.1

In some cases, the developers have built source code which has many defects. In fact, the defects are found to be more than the average number of defects expected for similar
components. In such cases, it definitely will take more time than planned to test such pieces of software. There could be test strategies deployed in such cases like:

- Use of time buffer for testing such pieces of software
- Use of additional testers for testing such pieces of software
- Use of more experienced testers on such pieces of software so that the testing velocity can be increased
- Use of overtime

However the best solution is to ensure that such cases do not happen. How to ensure that? Using quality assurance policies and making sure all developers and other software engineers stick to quality parameters. If quality norms are adhered to from the very inception of the project and all tasks on the project are delivered after strict compliance with quality norms then too many defects creeping into work products is not possible.

Answer 7.2

There are some interesting phenomena which can happen on a project. One such phenomenon is the actual speed of a project team different from any other project team for doing similar type of project work. If there are 2 teams on the same project developing different pieces of software product then definitely speed of one team will never be the same with which the other team will be developing pieces of the software product. Over time, the work completion dates will vary from projected dates. If a project team is slower in delivering the work products compared to projected dates then the project manager should work with that team to ensure that they deliver on time in future project work. If on the other hand a project team delivers its work products faster than scheduled then the project manager should first check the project plan to ensure that the project plan is correct. If the project plan is correct and the team delivers faster than expected then the project manager should reward them for their excellent effort. For future work, the project manager should take into account the speed of each team in delivering work products. The pay rate should also be adjusted keeping in mind the speed or the productivity.

Chapter 8

Answer 8.1

On iterative projects, project iteration closure happens many times. First project iterations close one after another in the project and finally the project itself closes. For example suppose 6 iterations were planned inside a project. Then the first iteration closes and only then the second iteration starts. The project itself will close after the 6th iteration closes. On iterative projects, since there is not much documentation apart from creating user manuals, documentation effort is not much during project closure on these projects. Also lessons learned will also not be part of project closure as no documentation has been created on the project.

On traditional projects, since there are no iterations, the project will close after all work on the project gets completed. Since on traditional projects, documentation forms a large part of
Software Project Management: A Process Driven Approach

total effort, ensuring closure of all kinds of documentation is essential activity on these projects. Lessons learnt is also a required part on these projects.

Answer 8.2

The foremost task at the closure of a project is to deliver all the deliverables including the software product made and user documents. The software product should be implemented at customer site and the project team should check if it is working properly. The project team should provide user training and hand over user manuals to end users.

At project completion, the project team is disbanded so that team members can work on next projects of the company. All the hardware and infrastructure used for the project should also be handed over to the company so that they can be assigned and used by other projects.

The project team also needs to perform some brainstorming to create lessons learnt on the project. The lessons learnt are very important documents to be used by next projects so that the mistakes done on the project are not repeated on future projects.

The project manager also creates project metrics to show performance of the project to the management.

Chapter 9

Answer 9.1

Most open source software development projects adopt some form of Agile methodology. At the beginning a few software developers gather and start the project. They build a small code base in form of a working prototype or a small software product to show to the world what they intend to do. If this product seems interesting and catches fancy of other developers then they join. At this stage some form of project management is initiated and work allocation to different team members is done. The project team may keep growing and they will keep building the product incrementally. They earn revenue through either providing support to users of the product or through providing end user training for the product.

Essentially most open source projects follow the incremental product development using any agile life cycle methodology.

Some examples of very successful open source projects include Apache, Google Chrome, Linux, mySQL etc.

Answer 9.2

The software product companies who build proprietary software products do business by selling their software products in the market. They keep their software source code secret so
that competitors can not copy their products and make similar products on their own. To develop such products requires the company to invest in starting projects to build these proprietary products. They hire project team members and initiate the project.

Companies or people who start an open source project keep their source code open to public so that other people who are interested in building these software product can join them or take the source code and build some product of their own. They earn money from providing support or training for these products to customers who like to use their products.

On these open source projects, fixing a release date for software product is difficult. It is because it is never known when developers may join the project and how much time they can devote on the project. Everything on the project is on voluntary basis in contrast to product development for proprietary software products where the owner fixes the release dates. The owner on these projects (for building proprietary products) has absolute control on the project. On open source projects there are no owners. There are only initiators. These initiators do not have much control on the project. But the good thing about open source projects is that best minds can join on the project and can build best quality software products. They do not have to abide by the dictats or fancy of any owner. So best minds can apply their creativity without any hindrance in developing such product.

Due to above mentioned reasons incremental agile methods work best for open source projects. Some examples include Linux operating system, Apache web server, mySQL database etc.