**Process assessment and Process Improvement of agile methodologies**

**Advanced Software Process (CPSC-­544).**

**Prof. Chang­Hyun Jo.**

**California State University, Fullerton**

**(Team-04)**

**Chirag Padsala. (**[**cpadsala@csu.fullerton.edu**](mailto:cpadsala@csu.fullerton.edu)**)**

**Prathmesh Pardhiye. (prathmesh@csu.fullerton.edu)**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Reviewer | Summary |
| Oct 23 | Both team members | Both team members | Discussion on mapping |
| Nov 4 | Both team members | Both team members | Discussion on gapping |
| Nov 5 | Both team members | Both team members | Discussion on Rating |
| Nov 10 | Prathmesh | Chirag | Introduction |
| Nov 12 | Prathmesh | Chirag | Examine business goals |
| Nov 14 | Chirag | Prathmesh | Initiate Process Improvement Cycle |
| Nov 15 | Chirag | Prathmesh | Performing Mapping |
| Nov 16 | Prathmesh | Chirag | Performing Gapping |
| Nov 19 | Prathmesh | Chirag | Rate each process attribute |
| Nov 21 | Chirag | Prathmesh | Develop action plan |
| Nov 23 | Both team members | Both team members | Step 5, Step 6, Step 7 and Step 8 |
| Nov 26 | Both team members | Both team members | Lessons Learned |
| Nov 27 | Both team members | Both team members | Team charter team evaluation |
| Nov 28 and Nov 29 | Both team members | Both team members | Report evaluation |

**Index**

Introduction…………………………………………………………………………………..4

The Steps involved in ISO-15504………………………………………………………….5

Step 1- Examine the organization business goals…………...……………………….....5

Step 2-Initiate process improvement cycle………...………………………………….…6

Step 3-To assess the current capability…………………………………………………..7

Mapping of Scrum XP to ISO 12207……………………………………………………..8

Gapping of current process to ISO12207……………………………………………...46

Rate process attribute……………………………………………………………………54

Step 4-Develop action plan……………………………………………………………….59

Identifying improvement areas………………………………………………………….59

Analyze assessment strengths and weaknesses…………………………………….59

Review Organizational improvement objectives………………………………………59

Analyze effectiveness measurements………………………………………………….61

Derive action plan………………………………………………………………………...61

Step 5 – Implement improvements Lessons……………………………………………62

Step 6 – Confirm improvements………………………………………………………….62

Step 7 – Sustain improvements…………………………………………………………..62

Step 8 – Monitor performance....…………………………………………………………62

Lesson Learned…………………………………………………………………………….63

Team Charter……………………………………………………………………………….66

Team Evaluation…………………………………………………………………………...70

**List of Figure**

Steps of Process Improvement ……………………………………..…………………...04

Process Determination Level….…………………………………………………………….58

**List of Tables**

Mapping for agile process to ISO 12207 ……..………………………………………….....08

Gaps between Scrum and XP process to ISO 12207…………….………………………...47

Process Attribute Rating and Capability Rating ……………………….…………………...54

Capability level and Process attributes ….……………………………...………………….58

Target and Assessed Process profiles ………………………………..…………………...58

**Introduction**

In this project, we performed process assessment which is ISO 12207 and process improvement plan ISO 15504 for our insertion sort software product which was developed in Android studio. For developing insertion sort application we followed scrum and XP which are agile processes for software development.

The objective of the process assessment and process improvement was to first map our scrum and XP process with the ISO 12207 processes. There were around 43 processes which were divided in terms of Software specific and system specific. All the processes were compared to the define process and then were mapped on to the ISO12207 process assessment for gapping. The mapping would then help to find the gaps between your current process and ISO 12207 processes. This would guide us to identify the necessary process required to develop our software product in a better and efficient way. The gaps helped us to rate each process attributes to identify the CMMI maturity level. The aim of the rating was to attain CMMI level 2 which is the managed process capability. The process attributes (PA) helped us to rate capability level of each process. The ISO15504 process improvement helped us to attain action plan for our current scum and XP process which would guide us to attain higher capability level.

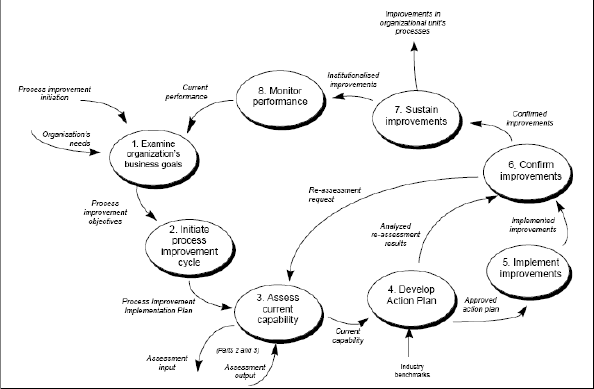


Figure: Steps of Process Improvement

The figure above illustrates the necessary steps which were considered to improve the current software process. There were various procedures which were required in the process improvement. The most important and the initial step is the examination of our organizations business goals. This helped us to focus on the necessary goals of the software product. Business goals helped us to start our process improvement cycle which guided us to attain our program plan. The next necessary step was to assess the current capability of our project to know capability level of our software product. The CMMI level then helped us to develop necessary action plan which would help us to attain the next capability level. The action plan which is developed is then approved by the assessors and the product owner to implement the improvements. This is called as the confirmation of the requirements which are then applied on to the software product for improvement. The improvement was then performed in organizational unit processes for institutionalised improvements. Then the monitoring of the performance is then done for the current process. Process improvement must be performed at regular intervals for efficient development and functioning of the software product. The required steps performed for process improvement of our insertion sort application has been listed and explained below in detail.

Steps for process improvement:

* Examine organization‘s business goals
* Initiate process improvement cycle
* Assess current capability
* Develop action plan
* Implement improvements
* Confirm improvements
* Sustain improvements
* Monitor performance

**Step 1: Examine Organization’s business goals**

We identified the business plan for our software. The necessary goal of our business plan was to:

* to achieve higher level of customer satisfaction
* achieve greater competitiveness
* achieve improved business value associated with the delivery of our software development services

The primary goal of this assessment was to check whether the current (Scrum and XP) process meets the ISO 12207 compliance for the perspective customers and to build an improvement plan to mediate any gaps. The customer capability level 2 was required to be attained for Software Requirements Analysis Process and SW Construction Process in the Software Implementation Processes Group and 3 processes in the Software Quality Assurance Process, Software Verification Process and Software Review Process which are Software Support Processes Group. The long term goal was to improve entire development process by having more mature documented process.

The key management problems become drivers that guide us to initiate process improvement throughout the organization with its objective. To attain the necessary goals the most important part was the customer satisfaction so that user gets the required product which is needed. The team’s development started with the user stories which were the user requirement for that product which helped us to attain the highest level of customer satisfaction. Reviews of the product were then taken so that the users get the desired software product. The assessment helped us to find the goals of the product and to improve the products quality and satisfy customer needs.

There are various software applications available in the various online application stores. The software product must be unique compared to all the other software applications that are available in the market. As there is competitiveness in the market so we developed a software product according to the customers need. If the needs of the customers are satisfied then it simultaneously helps to develop a successful product. This helped us to achieve greater competiveness between different software products.

The software product was developed in the form of iteration. The iteration format was an efficient software development tool because it guided us to make changes in the software product for the next iteration which were based on the customers and product owners need. The business goal helped us to find and improve gaps between the current processes to the process assessment standards. The organizations business goals were essential to build executive awareness of the necessity for a process improvement program. From the analysis of the organizations business goals and existing stimuli for improvement guided us in initiating process improvement cycle.

**Step 2: Initiate Process Improvement Cycle**

The process improvement program was implemented in our program by defining the following components:

* key roles and responsibilities
* project management
* budget
* milestones and
* accountability
* sponsorship
* background
* risks

The purpose of process initiation is to ensure that the improvement tasks are performed as a legitimate part in the organization. The process improvement for software product development was lead by the program manager. The software project sponsor was identified to provide funding for various resources and ensure participation from the organization. The project sponsor was provided with sufficient amount of authority and funds. The monitoring of the process improvement was performed by specifically formed assessment team. The engineering team performed various task in the field of development, quality assurance and project management. The assessment team held continuous meeting with the project manager, engineering team and other member for process improvement.

The goal of the process improvement cycle was help to assess the current Scrum and XP process with the ISO15504 for process improvement and find critical missing information in the process. Solving various critical processes helped us in the project develop. The budget was one of the most important factor which was considered in our process improvement. Estimates were carried out for allocating, resources, expenses and cost was figured out. Key roles were then assigned and responsibilities were given to all the team.

The program involved 6 months iterative improvement cycle with document review for investors to evaluation. The assessment team performed assessment by comparing the current Scum and XP processes with the ISO 12207 for finding the necessary processes which were missing in the current process standards. Our process improvement program would develop ISO 15504 standard form international organization for standardization. The project management handled the status of the process improvement and the necessary processes necessary for improvement. The risks regarding the improvement were also evaluated by considering the expected faults and error prior to performing assessment. The assessment team would keep track of improvement and document all the information which was found during assessment. The process improvement plan involved mapping the current process with the ISO 12007 process. This helped us to find the gaps between the Scrum and XP process with the standardized process. The gaping then guided us to rate the all the processes which helped to understand the CMMI level of the software product. Action plan was then performed for improving the capability level of the application. The process improvement was to find the critical processes and to assess those processes to improve efficiency and increase productivity.

**Step 3: Assess current capability**

The assessment was performed to identity the current level of the software and find gaps that were left to reach the specific capability level. The aim of the project was to reach level 2 of the CMMI model. To achieve this goal, we had to map our current Scrum and XP process with the ISO12207 process assessment model. Then the mapping helped us to find the gaps in our current process compared to ISO 12207 process assessment model. The gaps guided us to find the missing rating for our current process. After identification of each rating, it helped us to find the CMMI level of our current software product. The steps to assess current capability are explained below:

* Mapping of current process to ISO 12207 process assessment model.
* Gapping of current process to ISO 12207 process assessment model.
* Rate each process attribute
* Rate a capability level

**Mapping of Scrum and XP process to ISO 12207 process assessment model**

The purpose was mapping was to compare the current process to the ISO 12207 process assessment model. In this there were around 43 processes which divided in two major sub divisions which was System Context Processes and Software Specific Processes. All the current processes were mapped according to the ISO 12207 processes which would help in finding the necessary gaps in our software product. The mapping of the process was explained the table below.

Table: Mapping for agile process to ISO 12207

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mapping Template for the Agile Process (Scrum &XP) to ISO 12207** | | | | |
| **Clauses in ISO 12207** | **Processes**  **in ISO 12207** | **Purpose and Outcomes**  **[AND] Activities**  **in ISO 12207** | **Agile Process: Scrum &XP Practices** |  |
| **6.1** | **Agreement Processes** |  |  |  |
| **6.1.1** | **Acquisition Process** | **6.1.1.1 Purpose:**  The purpose of the Acquisition Process is to obtain the product and/or service that satisfy the need expressed by the acquirer. The process begins with the identification of customer needs and ends with the acceptance of the product and/or service needed by the acquirer.  **6.1.1.2 Outcomes:**  As a result of successful implementation of the Acquisition Process:  a) acquisition needs, goals, product and/or service acceptance criteria and acquisition strategies are  defined;  b) an agreement is developed that clearly expresses the expectation, responsibilities and liabilities of both  the acquirer and the supplier;  c) one or more suppliers is selected;  d) a product and/or service is acquired that satisfies the acquirer’s stated need;  e) the acquisition is monitored so that specified constraints such as cost, schedule and quality are met;  f) supplier deliverables are accepted; and  g) any identified open items have a satisfactory conclusion as agreed to by the acquirer and the supplier. | User stories were created to obtain the product needs that satisfy the requirements given by the customers and the product owner.  Product backlog was created based on the users stories that were implemented depending on the requirements. The necessary user stories were selected for development and were prioritized according to the needs. Schedule column was added onto the product backlog for each task and monitoring was done by the scrum master. |  |
| **6.1.1.3.1** |  | **Acquisition preparation** | In pre game planning and sprint planning acquisition preparation was performed |  |
| **6.1.1.3.2** |  | **Acquisition advertisement** | No mapping found |  |
| **6.1.1.3.3** |  | **Supplier selection** | No mapping found |  |
| **6.1.1.3.4** |  | **Contract agreement** | Product backlog was the agreement for all the required task |  |
| **6.1.1.3.5** |  | **Agreement monitoring** | Scrum master performed the monitoring of the agreement |  |
| **6.1.1.3.6** |  | **Acquirer acceptance** | Acceptance test was performed for acceptance |  |
| **6.1.1.3.7** |  | **Closure** | No mapping found |  |
| **6.1.2** | **Supply Process** | **6.1.2.1 Purpose**  The purpose of the Supply Process is to provide a product or service to the acquirer that meets the agreed requirements.  **6.1.2.2 Outcomes**  As a result of successful implementation of the Supply Process:  a) an acquirer for a product or service is identified;  b) a response to an acquirer's request is produced;  c) an agreement is established between the acquirer and the supplier for developing, maintaining, operating,  packaging, delivering, and installing the product and/or service;  d) a product and/or service that meets the agreed requirements are developed by the supplier;  e) the product and/or service is delivered to the acquirer in accordance with the agreed requirements; and  f) the product is installed in accordance with the agreed requirements. | Insertion sort application was developed based on the customer needs and was released in small iteration so that the missing requirements by the customer could be accomplished.  Product owner and the customers were the acquirer for the software application. User stories were created for acquirer needs .The necessary requirements were established on to the product backlog. The agreed requirements were developed in iteration and are made available to the acquirer in small releases. |  |
| **6.1.2.3.1** |  | **Opportunity identification** | In, user stories some of the opportunities were identified |  |
| **6.1.2.3.2** |  | **Supplier tendering** | No mapping found |  |
| **6.1.2.3.3** |  | **Contract agreement** | Product backlog was created for the requirement agreement |  |
| **6.1.2.3.4** |  | **Contract execution** | Contract execution was done by developing the software application according to the product backlog. |  |
| **6.1.2.3.5** |  | **Product/service delivery and support** | The product was delivered in small releases |  |
| **6.1.2.3.6** |  | **Closure** | No mapping found |  |
| **6.2** | **Organizational Project-Enabling Processes** |  |  |  |
| **6.2.1** | **Life Cycle Model Management Process** | **6.2.1.1 Purpose**  The purpose of the Life Cycle Model Management Process is to define, maintain, and assure availability of policies, life cycle processes, life cycle models, and procedures for use by the organization with respect to the scope of this International Standard.  This process provides life cycle policies, processes, and procedures that are consistent with the organization’s objectives, that are defined, adapted, improved and maintained to support individual project needs within the context of the organization, and that are capable of being applied using effective, proven methods and tools.  **6.2.1.2 Outcomes**  As a result of the successful implementation of the Life Cycle Model Management Process:  a) policies and procedures for the management and deployment of life cycle models and processes are  provided;  b) responsibility, accountability and authority for life cycle management are defined;  c) life cycle processes, models and procedures for use by the organization are defined, maintained and improved; and  d) prioritized process improvements are implemented. | In pre game planning the life cycle model was developed for the purpose to manage each iteration. Product backlog, user stories were the elements which were considered in the life cycle model.  Sprint planning and product backlog were the policies and procedures that were performed to manage the life cycle model. The responsibility, authority management was handled in pre game planning. The maintenance and improvement of life cycle was done in sprint backlog. The prioritization of implemented elements was performed by product owner in product backlog. |  |
| **6.2.1.3.1** |  | **Process establishment** | The establishment of the process is done in product backlog |  |
| **6.2.1.3.2** |  | **Process assessment** | For the assessment purpose the product backlog was compared. |  |
| **6.2.1.3.3** |  | **Process improvement** | Frequent refactoring was performed for improvement of process. |  |
| **6.2.2** | **Infrastructure Management Process** | **6.2.2.1 Purpose**  The purpose of the Infrastructure Management Process is to provide the enabling infrastructure and services to projects to support organization and project objectives throughout the life cycle.  This process defines, provides and maintains the facilities, tools, and communications and information  technology assets needed for the organization’s business with respect to the scope of this International  Standard.  **6.2.2.2 Outcomes**  As a result of the successful implementation of the Infrastructure Management Process:  a) the requirements for infrastructure to support processes are defined;  b) the infrastructure elements are identified and specified;  c) infrastructure elements are acquired;  d) the infrastructure elements are implemented; and  e) a stable and reliable infrastructure is maintained and improved.  NOTE The infrastructure elements may include hardware, software, methods, tools, techniques, standards, and facilities for development, operation, or maintenance. | Scrum and XP processes were performed to manage and develop the infrastructure of the software product.  For scrum and XP processes infrastructure elements like user stories and product backlog was implemented. These elements were developed in iterations and the scrum master made sure of the projects stability and reliability. |  |
| **6.2.2.3.1** |  | **Process implementation** | Pre game planning and staging for implementation process |  |
| **6.2.2.3.2** |  | **Establishment of the infrastructure** | Scrum and XP was the infrastructure of the project |  |
| **6.2.2.3.3** |  | **Maintenance of the infrastructure** | Scrum master handled the maintenance |  |
| **6.2.3** | **Project Portfolio Management Process** | **6.2.3.1 Purpose**  The purpose of the Project Portfolio Management Process is to initiate and sustain necessary, sufficient and suitable projects in order to meet the strategic objectives of the organization.  This process commits the investment of adequate organization funding and resources, and sanctions the authorities needed to establish selected projects. It performs continued qualification of projects to confirm they justify, or can be redirected to justify, continued investment.  **6.2.3.2 Outcomes**  As a result of the successful implementation of the Project Portfolio Management Process:  a) business venture opportunities, investments or necessities are qualified, prioritized and selected;  b) resources and budgets for each project are identified and allocated;  c) project management accountability and authorities are defined;  d) projects meeting agreement and stakeholder requirements are sustained; and  e) projects not meeting agreement or stakeholder requirements are redirected or terminated. | No mapping found |  |
| **6.2.3.3.1** |  | **Project initiation** | No mapping found |  |
| **6.2.3.3.2** |  | **Portfolio evaluation** | No mapping found |  |
| **6.2.3.3.3** |  | **Project closure** | No mapping found |  |
| **6.2.4** | **Human Resource Management Process** | **6.2.4.1 Purpose**  The purpose of the Human Resource Management Process is to provide the organization with necessary human resources and to maintain their competencies, consistent with business needs.  The process assures the providing of a supply of skilled and experienced personnel qualified to perform lifecycle processes to achieve organization, project and customer objectives.  **6.2.4.2 Outcomes**  As a result of the successful implementation of the Human Resource Management Process:  a) skills required by projects are identified;  b) necessary human resources are provided to projects;  c) skills of personnel are developed, maintained or enhanced;  d) conflicts in multi-project resource demands are resolved; and  e) individual knowledge, information and skills are collected, shared, reused and improved throughout the organization. | The human resources consisted of the product owner, scrum master and the development team. Each individual was divided according to their skills and experience.  The product owner, scum master, development team and customers were the necessary resources for our project. All the tasks were allocated according to the skills of each individual. |  |
| **6.2.4.3.1** |  | **Skill identification** | Skills of each individual were identified and allocated |  |
| **6.2.4.3.2** |  | **Skill development** | Android studio and scrum master were the necessary skills required for development. |  |
| **6.2.4.3.3** |  | **Skill acquisition and provision** | No mapping found |  |
| **6.2.4.3.4** |  | **Knowledge management** | Pair programming was performed to improve skill. |  |
| **6.2.5** | **Quality Management Process** | **6.2.5.1 Purpose**  The purpose of the Quality Management Process is to assure that products, services and implementations of  life cycle processes meet organizational quality objectives and achieve customer satisfaction.  **6.2.5.2 Outcomes**  As a result of the successful implementation of the Quality Management process:  a) organization quality management policies and procedures are defined;  b) organization quality objectives are defined;  c) accountability and authority for quality management are defined;  d) the status of customer satisfaction is monitored; and  e) appropriate action is taken when quality objectives are not achieved. | The scrum master assures that products and implementations of life cycle processes have met the organizational quality objectives and achieve customer satisfaction.  Daily scum meeting was performed during the development phase so that the defined procedures and policies were performed. The scrum master took note of the status of the project and if the quality of the project is achieved. |  |
| **6.2.5.3.1** |  | **Quality management** | Quality was maintained in daily scrum meeting and sprint review |  |
| **6.2.5.3.2** |  | **Quality management corrective action** | IF any non compliance was identified then corrective action were taken and implemented |  |
| **6.3** | **Project Processes** |  |  |  |
| **6.3.1** | **Project Planning** | **6.3.1.1 Purpose**  The purpose of the Project Planning Process is to produce and communicate effective and workable project plans.  This process determines the scope of the project management and technical activities, identifies process outputs, project tasks and deliverables, establishes schedules for project task conduct, including achievement criteria, and required resources to accomplish project tasks.  **6.3.1.2 Outcomes**  As a result of successful implementation of the Project Planning Process:  a) the scope of the work for the project is defined;  b) the feasibility of achieving the goals of the project with available resources and constraints are evaluated;  c) the tasks and resources necessary to complete the work are sized and estimated;  d) interfaces between elements in the project, and with other project and organizational units, are identified;  e) plans for the execution of the project are developed; and  f) plans for the execution of the project are activated. | The development of the project was done in iteration using scrum and XP process. Sprint planning and pre game planning was performed for implementing various task and resources required to accomplish project goal.  In pregame planning the scope of the project was identified. The necessary tasks and resources was sized and estimated in sprint planning during iteration. The execution of the project was done in small releases. |  |
| **6.3.1.3.1** |  | **Project initiation** | Initiation started with pre game planning |  |
| **6.3.1.3.2** |  | **Project planning** | Sprint Planning was performed for planning each iteration |  |
| **6.3.1.3.3** |  | **Project activation** | Development of each sprint was started after planning |  |
| **6.3.2** | **Project Assessment and Control Process** | **6.3.2.1 Purpose**  The purpose of the Project Assessment and Control Process is to determine the status of the project and ensure that the project performs according to plans and schedules, and within projected budgets, and that it satisfies technical objectives.  This process includes redirecting the project activities, as appropriate, to correct identified deviations and variations from other project management or technical processes. Redirection may include replanning as appropriate.  **6.3.2.2 Outcomes**  As a result of the successful implementation of the Project Assessment and Control Process:  a) progress of the project is monitored and reported;  b) interfaces between elements in the project, and with other project and organizational units, are monitored;  c) actions to correct deviations from the plan and to prevent recurrence of problems identified in the project  are taken when project targets are not achieved; and  d) project objectives are achieved and recorded. | The status of the project was monitored and controlled in the daily scrum meeting and sprint review which was performed by the scrum master.    Any deviations found are rectified in the next sprint review meeting. The deviations of the project were checked in each sprint planning and sprint review meeting. The monitoring was performed by the scrum master. The projects information was documented and the objectives were achieved |  |
| **6.3.2.3.1** |  | **Project monitoring** | Project monitoring was performed by scrum master |  |
| **6.3.2.3.2** |  | **Project control** | The project control was handled in sprint planning |  |
| **6.3.2.3.3** |  | **Project assessment** | No mapping found |  |
| **6.3.2.3.4** |  | **Project closure** | No mapping found |  |
| **6.3.3** | **Decision Management Process** | **6.3.3.1 Purpose**  The purpose of the Decision Management Process is to select the most beneficial course of project action where alternatives exist.  This process responds to a request for a decision encountered during the system life cycle, whatever its nature or source, in order to reach specified, desirable or optimized outcomes. Alternative actions reanalyzed and a course of action selected and directed. Decisions and their rationale are recorded to support future decision-making.  **6.3.3.2 Outcomes**  As a result of the successful implementation of the Decision Management Process:  a) a decision-making strategy is defined;  b) alternative courses of action are defined;  c) a preferred course of action is selected; and  d) the resolution, decision rationale and assumptions are captured and reported. | No mapping found |  |
| **6.3.3.3.1** |  | **Decision planning** | No mapping found |  |
| **6.3.3.3.2** |  | **Decision analysis** | No mapping found |  |
| **6.3.3.3.3** |  | **Decision tracking** | No mapping found |  |
| **6.3.4** | **Risk Management Process** | **6.3.4.1 Purpose**  The purpose of the Risk Management Process is to identify, analyze, treat and monitor the risks continuously.  The Risk Management Process is a continuous process for systematically addressing risk throughout the lifecycle of a system or software product or service. It can be applied to risks related to the acquisition, development, maintenance or operation of a system.  **6.3.4.2 Outcomes**  As a result of successful implementation of the Risk Management Process:  a) the scope of risk management to be performed is determined;  b) appropriate risk management strategies are defined and implemented;  c) risks are identified as they develop and during the conduct of the project;  d) risks are analyzed, and the priority in which to apply resources to treatment of these risks is determined;  e) risk measures are defined, applied, and assessed to determine changes in the status of risk and the  progress of the treatment activities; and  f) appropriate treatment is taken to correct or avoid the impact of risk based on its priority, probability, and consequence or other defined risk threshold. | Architectural influential factor was implemented before the development of the project for managing expected risks during operation, development and maintenance of the system.  The various forms of risk like cost, modification and security were major forms of risk for the project which were analyzed before the development and necessary steps were performed to avoid such risks. The risky part of the project while developing the code was monitored and developed in the presence of team members in scrum meetings. |  |
| **6.3.4.3.1** |  | **Risk management planning** | Risk management planning was performed during pre game planning |  |
| **6.3.4.3.2** |  | **Risk profile management** | Architectural influential factor |  |
| **6.3.4.3.3** |  | **Risk analysis** | The project is analyzed to find the risk and its impact on project. |  |
| **6.3.4.3.4** |  | **Risk treatment** | These risks were treated by their priority. |  |
| **6.3.4.3.5** |  | **Risk monitoring** | Risks were monitored periodically. |  |
| **6.3.4.3.6** |  | **Risk management process evaluation** | No mapping found |  |
| **6.3.5** | **Configuration Management Process** | **6.3.5.1 Purpose**  The purpose of the Configuration Management Process is to establish and maintain the integrity of all identified outputs of a project or process and make them available to concerned parties.  **6.3.5.2 Outcomes**  As a result of the successful implementation of the Configuration Management Process:  a) a configuration management strategy is defined;  b) items requiring configuration management are defined;  c) configuration baselines are established;  d) changes to items under configuration management are controlled;  e) the configuration of released items is controlled; and  f) the status of items under configuration management is made available throughout the life cycle. | Sprint planning was performed to maintain the integrity of the defined outputs and the product was released in small release to make them available to the concerned parties.  The configuration management strategy was defined in pre game planning. The management of the software product was handled by the product backlog. Continues integration was performed to configure release items. The status of item is made available in the product backlog |  |
| **6.3.5.3.1** |  | **Configuration management planning** | The configuration management planning was performed in pre game planning |  |
| **6.3.5.3.2** |  | **Configuration management execution** | The product backlog handled the configuration management execution |  |
| **6.3.6** | **Information Management Process** | **6.3.6.1 Purpose**  The purpose of the Information Management Process is to provide relevant, timely, complete, valid and, if required, confidential information to designated parties during and, as appropriate, after the system life cycle.  This process generates, collects, transforms, retains, retrieves, disseminates and disposes of information. It manages designated information, including technical, project, organizational, agreement and user information.  **6.3.6.2 Outcomes**  As a result of the successful implementation of the Information Management Process:  a) information to be managed is identified;  b) the forms of the information representations are defined;  c) information is transformed and disposed of as required;  d) the status of information is recorded;  e) information is current, complete and valid; and  f) information is made available to designated parties.  NOTE The Software Documentation Management Process is a specialization of the Information Management  Process and is included in the Software Support Process Group. | Daily scrum meeting and sprint planning were held to provide necessary information to be managed and to share information among different teams.  Product backlog was created during pre game planning for all the tasks to be done for development and sprint backlog was created in sprint planning and updated according to the process of development. |  |
| **6.3.6.3.1** |  | **Information management planning** | Pre game planning and Sprint planning |  |
| **6.3.6.3.2** |  | **Information management execution** | Execution task from product backlog according to their priorities and schedule. |  |
| **6.3.7** | **Measurement Process** | **6.3.7.1 Purpose**  The purpose of the Measurement Process is to collect, analyze, and report data relating to the products developed and processes implemented within the organizational unit, to support effective management of the processes, and to objectively demonstrate the quality of the products.  **6.3.7.2 Outcomes**  As a result of successful implementation of the Measurement Process:  a) the information needs of technical and management processes are identified;  b) an appropriate set of measures, driven by the information needs are identified and/or developed;  c) measurement activities are identified and planned;  d) the required data are collected, stored, analyzed, and the results interpreted;  e) information products are used to support decisions and provide an objective basis for communication;  f) the Measurement Process and measures are evaluated; and  g) improvements are communicated to the Measurement Process owner. | Sprint review where held at the end of each iteration. In this information implemented during development was collected, reported and analyzed for effective management and improve quality of the product.  The information which is collected during sprint review meeting is analyzed according to the product backlog and evaluation was done for the next iteration. |  |
| **6.3.7.3.1** |  | **Measurement planning** | Sprint review meeting for the next iteration |  |
| **6.3.7.3.2** |  | **Measurement performance** | Product backlog was measured for performance |  |
| **6.3.7.3.3** |  | **Measurement evaluation** | Measurement Evaluation was performed in the next Sprint Planning |  |
| **6.4** | **Technical Processes** |  |  |  |
| **6.4.1** | **Stakeholder Requirements Definition Process** | **6.4.1.1 Purpose**  The purpose of the Stakeholder Requirements Definition Process is to define the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment.  It identifies stakeholders, or stakeholder classes, involved with the system throughout its life cycle, and their needs and desires. It analyzes and transforms these into a common set of stakeholder requirements that express the intended interaction the system will have with its operational environment and that are the reference against which each resulting operational service is validated in order to confirm that the system fulfils needs.  **6.4.1.2 Outcomes**  As a result of successful implementation of the Stakeholder Requirements Definition Process:  a) the required characteristics and context of use of services are specified;  b) the constraints on a system solution are defined;  c) traceability of stakeholder requirements to stakeholders and their needs is achieved;  d) the basis for defining the system requirements is described;  e) the basis for validating the conformance of the services is defined; and  f) a basis for negotiating and agreeing to supply a service or product is provided. | User stories were created for the requirements of the software product in pre game planning. Product backlog was created by the selected user stores.  In pre game planning user stories were given by the product owner and customers. The selected user stories were then added onto the product backlog and were prioritized by the product owner. |  |
| **6.4.1.3.1** |  | **Stakeholder identification** | Product owner was our stakeholder |  |
| **6.4.1.3.2** |  | **Requirements identification** | User stories were created for requirements |  |
| **6.4.1.3.3** |  | **Requirements evaluation** | Product Backlog was created for evaluation |  |
| **6.4.1.3.4** |  | **Requirements agreement** | In pre game planning all the requirements were agreed. |  |
| **6.4.1.3.5** |  | **Requirement recording** | Requirements were recorded no to the product backlog |  |
| **6.4.2** | **System Requirements Analysis Process** | **6.4.2.1 Purpose**  The purpose of System Requirements Analysis is to transform the defined stakeholder requirements into a set  of desired system technical requirements that will guide the design of the system.  **6.4.2.2 Outcomes**  As a result of successful implementation of System Requirements Analysis:  a) a defined set of system functional and non-functional requirements describing the problem to be solved  are established;  b) the appropriate techniques are performed to optimize the preferred project solution;  c) system requirements are analyzed for correctness and testability;  d) the impact of the system requirements on the operating environment are understood;  e) the requirements are prioritized, approved and updated as needed;  f) consistency and traceability are established between the system requirements and the customer’s  requirements baseline;  g) changes to the baseline are evaluated for cost, schedule and technical impact; and  h) the system requirements are communicated to all affected parties and baselined. | We performed requirement analysis by creating the product backlog.  Requirements were identified as Functional and non functional requirement. Consistency and availability check were analyzed during scrum meeting. |  |
| **6.4.2.3.1** |  | **Requirements specification** | User stories were created by users and product owner |  |
| **6.4.2.3.2** |  | **Requirements evaluation** | Product backlog was created using users stories |  |
| **6.4.3** | **System Architectural Design Process** | **6.4.3.1 Purpose**  The purpose of the System Architectural Design Process is to identify which system requirements should be allocated to which elements of the system.  **6.4.3.2 Outcomes**  As a result of successful implementation of the System Architectural Design Process:  a) a system architecture design is defined that identifies the elements of the system and meets the defined  requirements;  b) the system’s functional and non-functional requirements are addressed;  c) the requirements are allocated to the elements of the system;  d) internal and external interfaces of each system element are defined;  e) verification between the system requirements and the system architecture is performed;  f) the requirements allocated to the system elements and their interfaces are traceable to the customer’s  requirements baseline;  g) consistency and traceability between the system requirements and system architecture design is  maintained; and  h) the system requirements, the system architecture design, and their relationships are baselined and  communicated to all affected parties;  I) human factors and ergonomic knowledge and techniques are incorporated in system design; and  j) human-centered design activities are identified and performed. | Requirements were mapped with work products during game planning.  System architecture diagram was created for defining elements. Functional and Non functional requirements were created for system requirements. Mapping of the required elements were added on to the product backlog. System context diagram was created based on these elements. User requirements and system architecture diagram was analyzed. |  |
| **6.4.3.3.1** |  | **Establishing architecture** | System context diagram was created for architecture |  |
| **6.4.3.3.2** |  | **Architectural evaluation** | Product backlog and system architecture diagram were analyzed. |  |
| **6.4.4** | **Implementation Process** | **6.4.4.1 Purpose**  The purpose of the Implementation Process is to realize a specified system element.  (No specified outcomes in document) | Hardware, software and other items such as product backlog, coding etc were identified |  |
| **6.4.5** | **System Integration Process** | **6.4.5.1 Purpose**  The purpose of the System Integration Process is to integrate the system elements (including software items, hardware items, manual operations, and other systems, as necessary) to produce a complete system that will satisfy the system design and the customers’ expectations expressed in the system requirements.  **6.4.5.2 Outcomes**  As a result of successful implementation of the System Integration Process:  a) a strategy is developed to integrate the system according to the priorities of the system requirements;  b) criteria are developed to verify compliance with the system requirements allocated to the system  elements, including the interfaces between system elements;  c) the system integration is verified using the defined criteria;  d) a regression strategy is developed and applied for re-testing the system when changes are made;  e) consistency and traceability are established between the system design and the integrated system  elements;  f) an integrated system is constructed that demonstrates compliance with the system design; and  g) an integrated system is constructed that demonstrates that a complete set of usable deliverable system elements exists. | Scrum and XP practice, mapping operation such product and sprint backlog, code etc. were integrated as a system.  Iterative developed practice was used for integration. System architecture diagram was created to verify compliance with the system. System elements were compared to system architecture diagram for verification. Integrated system was constructed in compliance to system design |  |
| **6.4.5.3.1** |  | **Integration** | Continuous integration was performed for each unit. |  |
| **6.4.5.3.2** |  | **Test readiness** | In this the product was compared with release backlog and evaluated for readiness |  |
| **6.4.6** | **System Qualification Testing Process** | **6.4.6.1 Purpose**  The purpose of the Systems Qualification Testing Process is to ensure that the implementation of each system requirement is tested for compliance and that the system is ready for delivery.  **6.4.6.2 Outcomes**  As a result of successful implementation of Systems Qualification Testing Process:  a) criteria for evaluating compliance with system requirements are developed;  b) the integrated system is tested using the defined criteria;  c) test results are recorded; and  d) readiness of the system for delivery is assured. | Unit testing was performed during each iteration and followed by that integration testing was conducted combining all the units.  With the help of these testing results product compliance was tested before its release. |  |
| **6.4.6.3.1** |  | **Qualification testing** | Unit test and integration tests were performed as a part of qualification testing |  |
| **6.4.7** | **Software Installation Process** | **6.4.7.1 Purpose**  The purpose of the Software Installation Process is to install the software product that meets the agreed requirements in the target environment.  **6.4.7.2 Outcomes**  As a result of successful implementation of the Software Installation Process:  a) a software installation strategy is developed;  b) criteria for software installation are developed that demonstrate compliance with the software installation requirements;  c) the software product is installed in the target environment; and  d) readiness of the software product for use in its intended environment is assured. | No mapping found |  |
| **6.4.7.3.1** |  | **Software installation** | No mapping found |  |
| **6.4.8** | **Software Acceptance Support Process** | **6.4.8.1 Purpose**  The purpose of the Software Acceptance Support Process is to assist the acquirer to achieve confidence that the product meets requirements.  **6.4.8.2 Outcomes**  As a result of the successful implementation of the Software Acceptance Support Process:  a) the product is completed and delivered to the acquirer;  b) acquirer acceptance tests and reviews are supported;  c) the product is put into operation in the customers’ environment; and  d) problems detected during acceptance are identified and communicated to those responsible for resolution.  NOTE Incremental delivery would be in completed increments. | Acceptance test was performed on developed system at end of each iteration.  Mapping was done between acquirer’s requirement and systems functionality. Missed requirement were integrated into the next iteration. |  |
| **6.4.8.3.1** |  | **Software acceptance support.** | System was developed as desired and initial support was provided. |  |
| **6.4.9** | **Software Operation Process** | **6.4.9.1 Purpose**  The purpose of the Software Operation Process is to operate the software product in its intended environment and to provide support to the customers of the software product.  **6.4.9.2 Outcomes**  As a result of the successful implementation of the Software Operation Process:  a) an operation strategy is defined;  b) conditions for correct operation of the software in its intended environment are identified and evaluated;  c) the software is tested and determined to operate in its intended environment;  d) the software is operated in its intended environment; and  e) assistance and consultation is provided to the customers of the software product in accordance with the agreement. | The software operation process was performed by developing application   on android IDE.  Software strategy was defined and testing was conducted during each iteration. |  |
| **6.4.9.3.1** |  | **Preparation for operation** | Learning android IDE and pre game planning was done for operation preparation. |  |
| **6.4.9.3.2** |  | **Operation activation and check-out** | No mapping found |  |
| **6.4.9.3.3** |  | **Operational use** | Operational use of the project was to provide sorted integer. The application was developed using Scrum and XP practices |  |
| **6.4.9.3.4** |  | **Customer support** | No mapping found |  |
| **6.4.9.3.5** |  | **Operation problem resolution** | No mapping found |  |
| **6.4.10** | **Software Maintenance Process** | **6.4.10.1 Purpose**  The purpose of the Software Maintenance Process is to provide cost-effective support to a delivered software product.  NOTE Pre-Delivery Software Maintenance activities include planning for post-delivery operations, supportability, and logistics determination. Post-delivery activities include software modification and operational support, such as training or operating a help desk.  **6.4.10.2 Outcomes**  As a result of successful implementation of the Software Maintenance Process:  a) a maintenance strategy is developed to manage modification and migration of products according to the release strategy;  b) the impact of changes to the existing system on organization, operations or interfaces are identified;  c) affected system and software documentation is updated as needed;  d) modified products are developed with associated tests that demonstrate that requirements are not compromised;  e) product upgrades are migrated to the customer’s environment; and  f) the system software modification is communicated to all affected parties. | No mapping found |  |
| **6.4.10.3.1** |  | **Process implementation** | No mapping found. |  |
| **6.4.10.3.2** |  | **Problem and modification analysis** | No mapping found. |  |
| **6.4.10.3.3** |  | **Modification implementation** | No mapping found. |  |
| **6.4.10.3.4** |  | **Maintenance review/acceptance** | No mapping found. |  |
| **6.4.10.3.5** |  | **Migration** | No mapping found. |  |
| **6.4.11** | **Software Disposal Process** | **6.4.11.1 Purpose**  The purpose of the Software Disposal Process is to end the existence of a system’s software entity.  This process ends active support by the operation and maintenance organization, or deactivates, disassembles and removes the affected software products, consigning them to a final condition and leaving  the environment in an acceptable condition. This process destroys or stores system software elements and related products in a sound manner, in accordance with legislation, agreements, organizational constraints and stakeholder requirements. Where required, it maintains records that may be monitored.  NOTE The objective is to retire a system's existing software products or services while preserving the integrity of organizational operations.  **6.4.11.2 Outcomes**  As a result of successful implementation of the Software Disposal Process:  a) a software disposal strategy is defined;  b) disposal constraints are provided as inputs to requirements;  c) the system's software elements are destroyed or stored;  d) the environment is left in an agreed-upon state; and  e) records allowing knowledge retention of disposal actions and any analysis of long-term impacts are available. | No mapping found. |  |
| **6.4.11.3.1** |  | **Software disposal planning** | No mapping found. |  |
| **6.4.11.3.2** |  | **Software disposal execution** | No mapping found. |  |
| **7** | **Software Specific Processes** |  |  |  |
| **7.1** | **Software Implementation Processes** |  |  |  |
| **7.1.1** | **Software Implementation Process** | **7.1.1.1 Purpose**  The purpose of the Software Implementation Process is to produce a specified system element implemented as a software product or service.  This process transforms specified behavior, interfaces and implementation constraints into actions that create a system element implemented as a software product or service, otherwise known as a "software item."  This process results in a software item that satisfies architectural design requirements through verification and stakeholder requirements through validation.  **7.1.1.2 Outcomes**  As a result of the successful implementation of the Software Implementation Process:  a) an implementation strategy is defined;  b) implementation technology constraints on the design are identified;  c) a software item is realized; and  d) a software item is packaged and stored in accordance with an agreement for its supply. | In the pre game planning stage user stories were created for implementation as software elements. These user stories consisted of the requirement of the customer and product owner. Theses user stories were then mapped on to the product backlog as the necessary software elements for development.  The product backlog consists of necessary software elements required for development. These elements were defined and prioritized and added by the product owner. System features were created for necessary items and was released in small releases. |  |
| **7.1.1.3.1** |  | **Software implementation strategy** | Implementation strategy was developed in pre game planning |  |
| **7.1.2** | **Software Requirements Analysis Process** | **7.1.2.1 Purpose**  The purpose of Software Requirements Analysis Process is to establish the requirements of the software elements of the system.  **7.1.2.2 Outcomes**  As a result of successful implementation of the Software Requirements Analysis Process:  a) the requirements allocated to the software elements of the system and their interfaces are defined;  b) software requirements are analyzed for correctness and testability;  c) the impact of software requirements on the operating environment are understood;  d) consistency and traceability are established between the software requirements and system  requirements;  e) prioritization for implementing the software requirements is defined;  f) the software requirements are approved and updated as needed;  g) changes to the software requirements are evaluated for cost, schedule and technical impact; and  h) the software requirements are baselined and communicated to all affected parties. | In the development stage,  User stories were created for requirements and then arranged them in product backlog as software elements of the system.  System context diagram was created to define software elements and interfaces of the system. Product backlog helped with the correctness regarding requirements and test were held during development and also approve the requirement and prioritize it. Requirements were divided into function and non functional requirements for their operating environment. Scrum meeting was held for creating product backlog which helps to establish traceability between software requirements and system requirements. Changes in software requirements were evaluated by product backlog for cost and technical impact. Scrum meetings were held for base lining requirements and communication to all affected parties |  |
| **7.1.2.3.1** |  | **Software requirements analysis** | Software requirement was analyzed during planning and staging and sprint planning. |  |
| **7.1.3** | **Software Architectural Design Process** | **7.1.3.1 Purpose**  The purpose of the Software Architectural Design Process is to provide a design for the software that implement and can be verified against the requirements.  **7.1.3.2 Outcomes**  As a result of successful implementation of the Software Architectural Design Process:  a) a software architectural design is developed and baselined that describes the software items that will  implement the software requirements;  b) internal and external interfaces of each software item are defined; and  c) Consistency and traceability are established between software requirements and software design. | System context diagram was developed during pre game planning stage for the elements required for the functioning of the product.  System features were created for adding elements in the system context diagram. These elements were then baselined and divided in functional and non functional requirements. Product backlog and sprint planning was performed for traceability and consistency |  |
| **7.1.3.3.1** |  | **Software architectural design** | System context diagram was developed for architectural design |  |
| **7.1.4** | **Software Detailed Design Process** | **7.1.4.1 Purpose**  The purpose of the Software Detailed Design Process is to provide a design for the software that implements and can be verified against the requirements and the software architecture and is sufficiently detailed to permit coding and testing.  **7.1.4.2 Outcomes**  As a result of successful implementation of the Software Detailed Design Process:  a) a detailed design of each software component, describing the software units to be built, is developed;  b) external interfaces of each software unit are defined; and  c) Consistency and traceability are established between the detailed design and the requirements and architectural design. | No mapping found. |  |
| **7.1.4.3.1** |  | **Software detailed design** | No mapping found. |  |
| **7.1.5** | **Software Construction Process** | **7.1.5.1 Purpose**  The purpose of the Software Construction Process is to produce executable software units that properly reflect the software design.  **7.1.5.2 Outcomes**  As a result of successful implementation of Software Construction Process:  a) verification criteria are defined for all software units against their requirements;  b) software units defined by the design are produced;  c) consistency and traceability are established between software units and requirements and design; and  d) Verification of the software units against the requirements and the design is accomplished. | We developed and delivered the product in iterations/small releases.  Tests were performed to define verification criteria for all software units according their requirements. Pair programming was performed for designing software units  Product backlog was created for maintaining consistency between software units and requirements. Sprint planning was conducted for verifying software units with requirements and to accomplish the design. |  |
| **7.1.5.3.1** |  | **Software construction** | Pre game planning and sprint planning were conducted for software construction. |  |
| **7.1.6** | **Software Integration Process** | **7.1.6.1 Purpose**  The purpose of the Software Integration Process is to combine the software units and software components, producing integrated software items, consistent with the software design, that demonstrate that the functional and non-functional software requirements are satisfied on an equivalent or complete operational platform.  **7.1.6.2 Outcomes**  As a result of successful implementation of the Software Integration Process:  a) an integration strategy is developed for software units consistent with the software design and the prioritized software requirements;  b) verification criteria for software items are developed that ensure compliance with the software  requirements allocated to the items;  c) software items are verified using the defined criteria;  d) software items defined by the integration strategy are produced;  e) results of integration testing are recorded;  f) consistency and traceability are established between software design and software items; and  g) a regression strategy is developed and applied for re-verifying software items when a change in software units (including associated requirements, design and code) occur. | Each component developed in units forms and then were integrated using continuous integration  During the sprint planning phase each units were assigned in daily scrum meeting and then integrated in the end of the iterative development  Tests were performed to ensure integration compliance. Software standards were used for verifying criteria. Integration strategy was developed at the end of all units. Results are produced in small releases after integration. Scrum master handle consistency and traceability between design and item. Same class was created so that it can be reused in the next iteration. |  |
| **7.1.6.3.1** |  | **Software integration** | Continuous integration was implemented. |  |
| **7.1.7** | **Software Qualification Testing Process** | **7.1.7.1 Purpose**  The purpose of the Software Qualification Testing Process is to confirm that the integrated software product meets its defined requirements.  **7.1.7.2 Outcomes**  As a result of successful implementation of the Software Qualification Testing Process:  a) criteria for the integrated software is developed that demonstrates compliance with the software  requirements;  b) integrated software is verified using the defined criteria;  c) test results are recorded; and  d) a regression strategy is developed and applied for re-testing the integrated software when a change in software items is made. | Scrum master made sure that the developed software product met its defined requirements.  Product backlog was created that demonstrated the software requirements Daily scrum meeting were held for verification of defined criteria  Tests were performed and recorded. |  |
| **7.1.7.3.1** |  | **Software qualification testing** | Testing was performed for qualification testing |  |
| **7.2** | **Software Support Processes** |  |  |  |
| **7.2.1** | **Software Documentation Management Process** | **7.2.1.1 Purpose**  The purpose of the Software Documentation Management Process is to develop and maintain the recorded software information produced by a process.  **7.2.1.2 Outcomes**  As a result of successful implementation of the Software Documentation Management Process:  a) a strategy identifying the documentation to be produced during the life cycle of the software product or  service is developed;  b) the standards to be applied for the development of the software documentation are identified;  c) documentation to be produced by the process or project is identified;  d) the content and purpose of all documentation is specified, reviewed and approved;  e) documentation is developed and made available in accordance with identified standards; and  f) documentation is maintained in accordance with defined criteria. | In pre game planning, the product backlog and user stories were some of the documents that were developed and recorded for implementation in the development phase.  The strategy for documentation of the software product was performed in pre game planning. Necessary standards were considered while developing. Product backlog, user stories and many more were the documentation produced for the project. Sprint reviews were performed at the end of each iteration for maintenance. |  |
| **7.2.1.3.1** |  | **Process implementation** | The implementation of documentation was performed in pregame planning |  |
| **7.2.1.3.2** |  | **Design and development** | Product backlog was documented for design |  |
| **7.2.1.3.3** |  | **Production** | The production was performed based on product backlog |  |
| **7.2.1.3.4** |  | **Maintenance** | The maintenance and review was done with sprint review |  |
| **7.2.2** | **Software Configuration Management Process** | **7.2.2.1 Purpose**  The purpose of the Software Configuration Management Process is to establish and maintain the integrity of the software items of a process or project and make them available to concerned parties.  **7.2.2.2 Outcomes**  As a result of successful implementation of the Software Configuration Management Process:  a) a software configuration management strategy is developed;  b) items generated by the process or project are identified, defined and baselined;  c) modifications and releases of the items are controlled;  d) modifications and releases are made available to affected parties;  e) the status of the items and modifications are recorded and reported;  f) the completeness and consistency of the items is ensured; and  g) the storage, handling and delivery of the items are controlled. | User stories were created and software standards were maintained to make the product available to concerned parties.  Sprint planning handled software configuration strategy. Product backlog items were identified, defined and baselined in iterative development. For modification and release were controlled by small release. Product was made available to the customer in the form of small release. The status of the items and modifications are recorded and reported in daily scrum. |  |
| **7.2.2.3.1** |  | **Process implementation** | Process implementation was done in Iterative development. |  |
| **7.2.2.3.2** |  | **Configuration identification** | Configuration identification was conducted during Pre-game planning. |  |
| **7.2.2.3.3** |  | **Configuration control** | Configuration control was handled Scrum master |  |
| **7.2.2.3.4** |  | **Configuration status accounting** | Daily scrum meeting was held for auditing and accounting. |  |
| **7.2.2.3.5** |  | **Configuration evaluation** | Evaluation was done during Sprint planning. |  |
| **7.2.2.3.6** |  | **Release management and delivery** | Sprint review gives release management and delivery overview. |  |
| **7.2.3** | **Software Quality Assurance Process** | **7.2.3.1 Purpose**  The purpose of the Software Quality Assurance Process is to provide assurance that work products and processes comply with predefined provisions and plans.  **7.2.3.2 Outcomes**  As a result of successful implementation of the Software Quality Assurance Process:  a) a strategy for conducting quality assurance is developed;  b) evidence of software quality assurance is produced and maintained;  c) problems and/or non-conformance with requirements are identified and recorded; and  d) adherence of products, processes and activities to the applicable standards, procedures and requirements are verified. | Product backlog was created so that work products and processes comply with predefined provisions and plans  In daily scrum meeting, the scrum master made sure the quality of the product is maintained Product backlog and Sprint backlog was produced and maintained for software quality assurance. Sprint planning and testing was conducted for problems and non-conformance with requirements and recorded in product backlog. In sprint planning and daily scrum meeting, the scrum master made sure the adherence of product and all the processes were completed according to the standards, procedures and requirements |  |
| **7.2.3.3.1** |  | **Process implementation** | Pre-game planning was conducted for process implementation. |  |
| **7.2.3.3.2** |  | **Product assurance** | Product backlog was created for product assurance. |  |
| **7.2.3.3.3** |  | **Process assurance** | Scrum master guided and validated process assurance. |  |
| **7.2.3.3.4** |  | **Assurance of quality systems** | Sprint review helps with the quality assurance after every sprint. |  |
| **7.2.4** | **Software Verification Process** | **7.2.4.1 Purpose**  The purpose of the Software Verification Process is to confirm that each software work product and/or service of a process or project properly reflects the specified requirements.  **7.2.4.2 Outcomes**  As a result of successful implementation of the Software Verification Process:  a) a verification strategy is developed and implemented;  b) criteria for verification of all required software work products is identified;  c) required verification activities are performed;  d) defects are identified and recorded; and  e) results of the verification activities are made available to the customer and other involved parties. | In daily scrum meeting, the scrum master verified that each software work product i.e. product backlog reflects the specified requirements.  Daily scrum meetings were held for verification strategy. Verification criteria for all the software work products were identified in product backlog. Daily scrum meeting and sprint backlog were conducted for verifying activities. Architectural influential factors and tests were performed for identifying and recording defects. In sprint review the results of the verification activities were made available to everyone. |  |
| **7.2.4.3.1** |  | **Process implementation** | Process implementation was done with the help of sprint planning. |  |
| **7.2.4.3.2** |  | **Verification** | Scrum master use to do the verification during and after sprint. |  |
| **7.2.5** | **Software Validation Process** | **7.2.5.1 Purpose**  The purpose of the Software Validation Process is to confirm that the requirements for a specific intended use of the software work product are fulfilled.  **7.2.5.2 Outcomes**  As a result of successful implementation of the Software Validation Process:  a) a validation strategy is developed and implemented;  b) criteria for validation of all required work products are identified;  c) required validation activities are performed;  d) problems are identified and recorded;  e) evidence is provided that the software work products as developed are suitable for their intended use;  and  f) results of the validation activities are made available to the customer and other involved parties. | The product backlog is created to make sure that the intended use of the software work products is fulfilled.  Product backlog is created for validation. Scrum master assures that all the criteria for validation of all required work products are identified.  Iterative development is performed.  Architectural influential factors helped to identify problems. Product owner makes sure that the software work products as developed are suitable for their intended use. Validation activity is then made available in the form of short release. |  |
| **7.2.5.3.1** |  | **Process implementation** | In sprint development the validation process was implemented |  |
| **7.2.5.3.2** |  | **Validation** | Product backlog was used for validation |  |
| **7.2.6** | **Software Review Process** | **7.2.6.1 Purpose**  The purpose of the Software Review Process is to maintain a common understanding with the stakeholders of the progress against the objectives of the agreement and what should be done to help ensure development of product that satisfies the stakeholders. Software reviews are at both project management and technical levels and are held throughout the life of the project.  **7.2.6.2 Outcomes**  As a result of successful implementation of the Software Review Process:  a) management and technical reviews are held based on the needs of the project;  b) the status and products of an activity of a process are evaluated through review activities;  c) review results are made known to all affected parties;  d) action items resulting from reviews are tracked to closure; and  e) risks and problems are identified and recorded. | Sprint reviews were performed at the end of each iteration for review and to know what should be done in the next iteration to ensure the development of the product satisfies the stakeholder’s requirements.  Daily scrum meeting and test were held for technical review and sprint review for management review. Products activity was evaluated in daily scrum meeting by comparing tasks with the product backlog. Review results were made which were known in sprint review. Architectural influential factors were created and tests were performed for risks and problems. |  |
| **7.2.6.3.1** |  | **Process implementation** | Process implementation was conducted during sprint planning. |  |
| **7.2.6.3.2** |  | **Project Management Reviews** | Sprint Review was used for project management. |  |
| **7.2.6.3.3** |  | **Technical Reviews** | Sprint review was analyzed for technical reviews. |  |
| **7.2.7** | **Software Audit Process** | **7.2.7.1 Purpose**  The purpose of the Software Audit Process is to independently determine compliance of selected products and processes with the requirements, plans and agreement, as appropriate.  **7.2.7.2 Outcomes**  As a result of successful implementation of the Software Audit Process:  a) an audit strategy is developed and implemented;  b) compliance of selected software work products and/or services or processes with requirements, plans  and agreement is determined according to the audit strategy;  c) audits are conducted by an appropriate independent party; and  d) Problems detected during an audit are identified and communicated to those responsible for corrective action, and resolution. | No mapping found |  |
| **7.2.7.3.1** |  | **Process implementation** | No mapping found |  |
| **7.2.7.3.2** |  | **Software audit** | No mapping found |  |
| **7.2.8** | **Software Problem Resolution Process** | **7.2.8.1 Purpose**  The purpose of the Software Problem Resolution Process is to ensure that all discovered problems are identified, analyzed, managed and controlled to resolution.  **7.2.8.2 Outcomes**  As a result of successful implementation of the Software Problem Resolution Process:  a) a problem management strategy is developed;  b) problems are recorded, identified and classified;  c) problems are analyzed and assessed to identify acceptable solution(s);  d) problem resolution is implemented;  e) problems are tracked to closure; and  f) the status of all problems reported is known.  NOTE The Software Problem Resolution Process could be used or easily adapted to manage, track and control  software change requests. | Architectural influential factors were considered for identifying various software problems. These problems were analyzed before the software development.  The problems were identified development phase for efficient functioning of the software product. The problems related to cost, modification and security was analyzed. Various types of tests were performed during coding for handling various problems that arises. |  |
| **7.2.8.3.1** |  | **Process implementation** | Pre-game planning helps with process implementation and Testing were conducted during sprint. |  |
| **7.2.8.3.2** |  | **Problem resolution** | Architectural influential factors provide problems and risk factors. |  |
| **7.3** | **Software Reuse Processes** |  |  |  |
| **7.3.1** | **Domain Engineering Process** | **7.3.1.1 Purpose**  The purpose of the Domain Engineering Process is to develop and maintain domain models, domain architectures and assets for the domain.  **7.3.1.2 Outcomes**  As a result of successful implementation of the Domain Engineering Process:  a) the representation forms for the domain models and the domain architectures are selected;  b) the boundaries of the domain and its relationships to other domains are established;  c) a domain model that captures the essential common and different features, capabilities, concepts, and  functions in the domain are developed;  d) a domain architecture describing the family of systems within the domain, including their commonalities and variabilities, is developed;  e) assets belonging to the domain are specified;  f) assets belonging to the domain are acquired or developed and maintained throughout their life cycles; and  g) the domain models and architectures are maintained throughout their life cycles. | No mapping found. |  |
| **7.3.1.3.1** |  | **Process implementation** | No mapping found. |  |
| **7.3.1.3.2** |  | **Domain analysis** | No mapping found. |  |
| **7.3.1.3.3** |  | **Domain design** | No mapping found. |  |
| **7.3.1.3.4** |  | **Asset provision** | No mapping found. |  |
| **7.3.1.3.5** |  | **Asset maintenance** | No mapping found. |  |
| **7.3.2** | **Reuse Asset Management Process** | **7.3.2.1 Purpose**  The purpose of the Reuse Asset Management Process is to manage the life of reusable assets from conception to retirement.  **7.3.2.2 Outcomes**  As a result of successful implementation of the Reuse Asset Management Process:  a) an asset management strategy is documented;  b) an asset classification scheme is established;  c) criteria for asset acceptance, certification and retirement are defined;  d) an asset storage and retrieval mechanism is operated;  e) the use of assets is recorded;  f) changes to the assets are controlled, and  g) users of assets are notified of problems detected, modifications made, new versions created and deletion of assets from the storage and retrieval mechanism. | No mapping found. |  |
| **7.3.2.3.1** |  | **Process implementation** | No mapping found. |  |
| **7.3.2.3.2** |  | **Asset storage and retrieval definition** | No mapping found. |  |
| **7.3.2.3.3** |  | **Asset management and control** | No mapping found. |  |
| **7.3.3** | **Reuse Program Management Process** | **7.3.3.1 Purpose**  The purpose of the Reuse Program Management Process is to plan, establish, manage, control, and monitor an organization’s reuse program and to systematically exploit reuse opportunities.  **7.3.3.2 Outcomes**  As a result of successful implementation of Reuse Program Management Process:  a) the organization’s reuse strategy, including its purpose, scope, goals and objectives, is defined;  b) the domains for potential reuse opportunities are identified;  c) the organization’s systematic reuse capability is assessed;  d) the reuse potential of each domain is assessed;  e) reuse proposals are evaluated to ensure the reuse product is suitable for the proposed application;  f) the reuse strategy is implemented in the organization;  g) feedback, communication, and notification mechanisms that operate between affected parties are  established; and  h) the reuse program is monitored and evaluated.  NOTE The affected parties may include reuse program administrators, asset managers, domain engineers, developers, operators, and maintainers. | We used iterative development process in which the product was launched in small releases.  In sprint planning reuse strategy was defined The product was developed in different iteration. Sprint reviews were done after each iteration Sprint reviews helped us to get feedback and communication for product development. Monitoring and evaluation of the product was done by the scrum master |  |
| **7.3.3.3.1** |  | **Initiation** | Sprint planning was performed for process initiation. |  |
| **7.3.3.3.2** |  | **Domain identification** | No mapping found. |  |
| **7.3.3.3.3** |  | **Reuse assessment** | Iterative development was performed. |  |
| **7.3.3.3.4** |  | **Planning** | Pre-game planning was conducted. |  |
| **7.3.3.3.5** |  | **Execution and Control** | Scrum master monitors the execution and control. |  |
| **7.3.3.3.6** |  | **Review and evaluation** | Sprint review provides review and evaluation. |  |

**Gapping of current process to ISO 12207 process assessment model**

The purpose of gapping was to find the missing components in the current process compared to ISO 12207 process assessment model. Our current process was fully compatible with both Scrum and XP processes. We assessed our current process against ISO 12207, because your customer requires you us follow ISO 12207, so we needed to find what are the gaps you need to fill in to satisfy your customer’s need. We had to perform gapping for 5 processes which are Software Requirements Analysis Process and SW Construction Process in the Software Implementation Processes Group and 3 processes in the Software Quality Assurance Process, Software Verification Process and Software Review Process which are Software Support Processes Group.

The gapping between the current process and the ISO 12207 helped us to find the necessary missing contents in our process. The one of the missing content that was discovered while assessment was to develop a backlog graph for the development our project. This was one of the necessary content which was required for development. The backlog graph needs to be developed for the purpose to check if the particular task is completed in the given period of time. It also helps to analyze which of the task which were required to be completed on time and the one’s which was accepted if late. If a particular task is not completed in the given time then backlog graph can be useful to adjust with other tasks so that the software product is completed in the defined period of time. This was one of the important attribute which was missing while development. Another necessary aspect which was found while finding the gaps in assessment was the management review was not held. The management review was required during the sprint review for finding the necessary management issue in the software product. The gaps regarding the current process compared to ISO 12207 is explained in detail below:

Table: Gaps between Scrum and XP process to ISO 12207

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gaps between the Agile Process (Scrum & XP processes) to ISO 12207**  [Scope: only for the Software Implement Processes Group and the Software Support Processes Group] | | | | |
| **Clauses in ISO 12207** | **Processes**  **in ISO 12207** | **Purpose and Outcomes**  **[AND] Activities**  **in ISO 12207** | **Agile Process: Scrum &XP Practices.** | **Gaps between your process (Scrum & XP practices) and ISO 12207.** |
| **7** | **Software Specific Processes** |  |  |  |
| **7.1** | **Software Implementation Processes** |  |  |  |
| **7.1.2** | **Software Requirements Analysis Process** | **7.1.2.1 Purpose**  The purpose of Software Requirements Analysis Process is to establish the requirements of the software elements of the system.  **7.1.2.2 Outcomes**  As a result of successful implementation of the Software Requirements Analysis Process:  a) the requirements allocated to the software elements of the system and their interfaces are defined;  b) software requirements are analyzed for correctness and testability;  c) the impact of software requirements on the operating environment are understood;  d) consistency and traceability are established between the software requirements and system  requirements;  e) prioritization for implementing the software requirements is defined;  f) the software requirements are approved and updated as needed;  g) changes to the software requirements are evaluated for cost, schedule and technical impact; and  h) the software requirements are baselined and communicated to all affected parties. | In the development stage,  User stories were created for requirements and then arranged them in product backlog as software elements of the system.  System context diagram was created to define software elements and interfaces of the system. Product backlog helped with the correctness regarding requirements and test were held during development and also approve the requirement and prioritize it. Requirements were divided into function and non functional requirements for their operating environment. Scrum meeting was held for creating product backlog which helps to establish traceability between software requirements and system requirements. Changes in software requirements were evaluated by product backlog for cost and technical impact. Scrum meetings were held for base lining requirements and communication to all affected parties | No gaps found.  No gaps found.  No gaps found.  No gaps found.  No gaps found.  No gaps found.  No gaps found.  No gaps found.  No gaps found. |
| **7.1.2.3.1** |  | **Software requirements analysis** | Software requirement was analyzed during planning and staging and sprint planning. | No gaps found. |
| **7.1.5** | **Software Construction Process** | **7.1.5.1 Purpose**  The purpose of the Software Construction Process is to produce executable software units that properly reflect the software design.  **7.1.5.2 Outcomes**  As a result of successful implementation of Software Construction Process:  a) verification criteria are defined for all software units against their requirements;  b) software units defined by the design are produced;  c) consistency and traceability are established between software units and requirements and design; and  d) verification of the software units against the requirements and the design is accomplished. | We developed and delivered the product in iterations/small releases.  Tests were performed to define verification criteria for all software units according their requirements. Pair programming was performed for designing software units  Product backlog was created for maintaining consistency between software units and requirements. Sprint planning was conducted for verifying software units with requirements and to accomplish the design. | No gaps found.  No gaps found.  No gaps found.  Backlog graph needs to created.  No gaps found. |
| **7.1.5.3.1** |  | **Software construction** | Pre game planning and sprint planning were conducted for software construction. | No gaps found. |
| **7.2** | **Software Support Processes** |  |  |  |
| **7.2.3** | **Software Quality Assurance Process** | **7.2.3.1 Purpose**  **The purpose of the Software Quality Assurance Process is to provide assurance that work products and processes comply with predefined provisions and plans.**  **7.2.3.2 Outcomes**  **As a result of successful implementation of the Software Quality Assurance Process:**  **a) a strategy for conducting quality assurance is developed;**  **b) evidence of software quality assurance is produced and maintained;**  **c) problems and/or non-conformance with requirements are identified and recorded; and**  **d) Adherence of products, processes and activities to the applicable standards, procedures and requirements are verified.** | Product backlog was created so that work products and processes comply with predefined provisions and plans.  In daily scrum meeting, the scrum master made sure the quality of the product is maintained Product backlog and Sprint backlog was produced and maintained for software quality assurance. Sprint planning and testing was conducted for problems and non-conformance with requirements and recorded in product backlog. In sprint planning and daily scrum meeting, the scrum master made sure the adherence of product and all the processes were completed according to the standards, procedures and requirements | No gaps found.  No gaps found.  No gaps found.  No gaps found.  No gaps found. |
| **7.2.3.3.1** |  | **Process implementation** | Pre-game planning was conducted for process implementation. | No gaps found. |
| **7.2.3.3.2** |  | **Product assurance** | Product backlog was created for product assurance. | No gaps found. |
| **7.2.3.3.3** |  | **Process assurance** | Scrum master guided and validated process assurance. | No gaps found. |
| **7.2.3.3.4** |  | **Assurance of quality systems** | Sprint review helps with the quality assurance after every sprint. | No gaps found. |
| **7.2.4** | **Software Verification Process** | **7.2.4.1 Purpose**  **The purpose of the Software Verification Process is to confirm that each software work product and/or service of a process or project properly reflects the specified requirements.**  **7.2.4.2 Outcomes**  **As a result of successful implementation of the Software Verification Process:**  **a) a verification strategy is developed and implemented;**  **b) criteria for verification of all required software work products is identified;**  **c) required verification activities are performed;**  **d) defects are identified and recorded; and**  **e) results of the verification activities are made available to the customer and other involved parties.** | In daily scrum meeting, the scrum master verified that each software work product i.e. product backlog reflects the specified requirements.  Daily scrum meetings were held for verification strategy. Verification criteria for all the software work products were identified in product backlog. Daily scrum meeting and sprint backlog were conducted for verifying activities. Architectural influential factors and tests were performed for identifying and recording defects. In sprint review the results of the verification activities were made available to everyone. | No gaps found.  No gaps found.  No gaps found.  No gaps found.  No gaps found.  No gaps found. |
| **7.2.4.3.1** |  | **Process implementation** | Process implementation was done with the help of sprint planning. | No gaps found. |
| **7.2.4.3.2** |  | **Verification** | Scrum master use to do the verification during and after sprint. | No gaps found. |
| **7.2.6** | **Software Review Process** | **7.2.6.1 Purpose**  **The purpose of the Software Review Process is to maintain a common understanding with the stakeholders of the progress against the objectives of the agreement and what should be done to help ensure development of a product that satisfies the stakeholders. Software reviews are at both project management and technical levels and are held throughout the life of the project.**  **7.2.6.2 Outcomes**  **As a result of successful implementation of the Software Review Process:**  **a) management and technical reviews are held based on the needs of the project;**  **b) the status and products of an activity of a process are evaluated through review activities;**  **c) review results are made known to all affected parties;**  **d) action items resulting from reviews are tracked to closure; and**  **e) risks and problems are identified and recorded.** | Sprint reviews were performed at the end of each iteration for review and to know what should be done in the next iteration to ensure the development of the product satisfies the stakeholder’s requirements.  Daily scrum meeting and test were held for technical review and sprint review for management review. Products activity was evaluated in daily scrum meeting by comparing tasks with the product backlog. Review results were made which were known in sprint review. Architectural influential factors were created and tests were performed for risks and problems. | No gaps found.  Sprint review does not provide enough management reviews.  Backlog graph needs to be established for the status of the process.  No gaps found.  Backlog graph required.  No gaps found. |
| **7.2.6.3.1** |  | **Process implementation** | Process implementation was conducted during sprint planning. | No gaps found. |
| **7.2.6.3.2** |  | **Project Management Reviews** | Sprint Review was used for project management. | No gaps found. |
| **7.2.6.3.3** |  | **Technical Reviews** | Sprint reviews were analyzed for technical reviews. | No gaps found. |

**Rate each Process attribute**

The rating of each process attribute was to find the capability level of each process. The process attributes in each rating was compared to the current process. The aim of our project assessment was to reach level 2 of the CMMI model. In order to achieve this level, rating of each process of the ISO 15504 was needed to be performed. The process attributes consisted of attributes which started from PA 1.1, 2.1, 2.2 and so on. The objective of this software project was compare till PA 2.2 of the rating model. If the particular process reaches level 2.2 then capability level 2 is achieved for that process.

In the rating model we had to perform gapping for 5 processes which are Software Requirements Analysis Process and SW Construction Process in the Software Implementation Processes Group and 3 processes in the Software Quality Assurance Process, Software Verification Process and Software Review Process which are Software Support Processes Group. The process attributes were provided rating in terms of capability level. Process attributes ratings were provided rating in terms of not achieve, partially achieved, largely achieved and fully achieved. The software specific processes, 5 processes were rated and gaps in ratings were found. These processes were then compared with the process attribute of ISO 15504 and analyzed in terms of process performance, performance management and work product management. The software requirement analysis process attained its required capability model because it had the full achievement of that particular attribute. The processes which need improvements to reach level 2 of the capability model are software construction process and software review process. The software construction process did not full achieve the process performance 1.1 which is the necessary attribute to reach level 2 and even needs improvement in performance management and work product management. Similarly, software review performance needs to perform develop action plan for improvement in that process. The rating of the processes illustrated in the table below:

Table: Process Attribute Rating and Capability Rating

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Rating Process Attributes (PA) and Capability Levels (CL)**  [**Target Profile: CL 2**]  **Process Attributes Rating [N/P/L/F] and Capability Rating [CL0 – CL5]** | | | | | | |
| **Clauses in ISO 12207** | **Processes**  **in ISO 12207** | **Process Attributes (PAs) in ISO 15504** | **Required PA Rating for CL** | **Process Attribute Rating**  **[N/P/L/F]** | **Gaps Found in Rating** | **CL Rating**  **[CL0 – CL5]** |
|  |  |  |  |  |  |  |
| **7.1** | **Software Implementation Processes** |  |  |  |  |  |
| **7.1.2** | **Software Requirement Analysis Process** | **PA 2.2 Work Product Management** | F | F | No Gaps Found. | CL 2 |
| **PA 2.1 Performance management** | F | F | No Gaps Found. |
| **PA 1.1 Process Performance** | F | F | No Gaps Found. |
| **7.1.5** | **Software Construction Process** | **PA 2.2 Work Product Management** | F | P | Backlog graph was required for work product monitoring. | CL 1 |
| **PA 2.1 Performance management** | F | L | Backlog graph is required for performance monitoring of a process. |  |
| **PA 1.1 Process Performance** | F | L | Backlog graph is required for measuring the performance. |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **7.2** | **Software Support Processes** |  |  |  |  |  |
| **7.2.3** | **Software Quality Assurance Process** | **PA 2.2 Work Product Management** | F | F | No Gaps Found. | CL 2 |
| **PA 2.1 Performance management** | F | F | No Gaps Found. |
| **PA 1.1 Process Performance** | F | F | No Gaps Found. |
| **7.2.4** | **Software Verification Process** | **PA 2.2 Work Product Management** | F | F | No Gaps Found. | CL 2 |
| **PA 2.1 Performance management** | F | F | No Gaps Found. |
| **PA 1.1 Process Performance** | F | F | No Gaps Found. |
| **7.2.6** | **Software Review Process** | **PA 2.2 Work Product Management** | F | P | Requirements gathered by reviews for work product needs to be defined. | CL 1 |
| **PA 2.1 Performance management** | F | P | Objective for performance management needs to be defined by fetching proper management and technical reviews. |
| **PA 1.1 Process Performance** | F | L | Management reviews and backlog graph are required for software process, |

**Rate a capability level:**

When we check the result of the processes, we generally rate the process attributes using N, P, L, F i.e.; Not achieved, partially achieved, largely achieved, fully achieved respectively. If the PA 1.1 for a process is fully or largely achieved, then the process achieves capability level 1. Similarly, for process to be on capability level 2 its process attribute 1.1 needs to be fully achieved and PA 2.1 and 2.2 should be largely or fully achieved. The process cannot be rated as level 2 if the PA 1.1 for a process is not fully achieved.

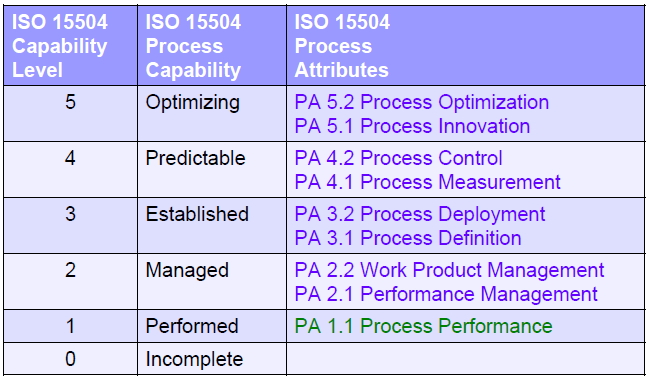
 As shown in the diagram below the process can achieve level 0-5. For a process to achieve level 4 all the below process attributes need to be fully achieved or else it cannot achieve the desired level.

Table: Capability level and Process attributes

For our process we used ISO 15504 for process assessment on the application which we developed using Scrum and XP. We did the mapping of System context process and software specific processes of ISO 12207 with our scrum and XP practices, followed by that we found gap using ISO 12207 which was useful for rating the process using process assessment in ISO 15504.

In our project we assessed 5 processes namely 7.1.2 Software requirement analysis process, 7.1.5 Software construction process, 7.2.3 Software quality assurance process 7.2.4 Software verification process and 7.2.5 Software review process. We found that 2 process were at capability level 1 and 3 process were at capability level 2. And for the process which were not able to achieve capability level 2 were considered as gap which needed to be fulfilled using the appropriate action plan.

The figure below shows the target and the assessed process profiles of all 5 process. For the Process 7.1.2 we had a target of all 3 process attribute to be fully achieved to achieve capability level 2. As there were no gaps found during mapping and gapping of this process we were able to rate this level as fully achieved and thus the process achieved capability level 2. Similarly, for the process 7.2.3 and 7.2.4 we were able to map all the practices with the ISO 12207 standards and no gaps were found. While for the process 7.1.5 and 7.2.6 we found gaps while mapping and gapping so these two process were not able to achieve capability level 2, and so we need to developed action plan for those process to improve the process and achieve the desired capability level 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Process | | Process Attributes | | |
| Performed | Managed | |
| PA 1.1 | PA 2.1 | PA 2.2 |
| 7.1.2 Software Requirement Analysis Process | Target | F | F | F |
| Assessed | F | F | F |
| 7.1.5 Software Construction Process | Target | F | F | F |
| Assessed | L | L | P |
| 7.2.3 Software Quality Assurance Process | Target | F | F | F |
| Assessed | F | F | F |
| 7.2.4 Software Verification Process | Target | F | F | F |
| Assessed | F | F | F |
| 7.2.6 Software Review Process | Target | F | F | F |
| Assessed | L | P | P |

Table: Target and Assessed Process profiles

The figure below shows the capability determination of the processes which were assessed using the ISO15504. It shows the bar graph representation of the capability level achieved by the process assessed. The X-axis represent process and Y-axis represent the capability level achieved by those process on X-axis. Maximum capability level which can be achieved by the process is 5 and minimum is 0. As per our assessment we needed our process to achieve level 2 or develop action plan for the process which did not achieve level 2.

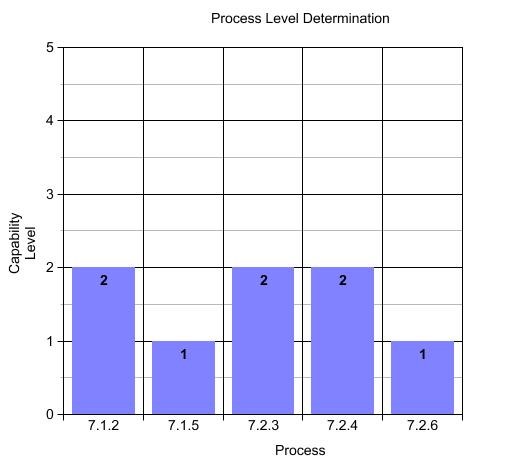


Figure: Process Determination Level

**Step 4- Develop action plan:**

The purpose of assessment was to analyze against the organizations business goals. The following steps were used to develop action plan. The steps are:

* Identify improvement areas.
* Analyze assessments strengths and weaknesses.
* Review organizational improvement objectives.
* Analyze effectiveness measurements.
* List improvement areas
* Derive action plan.

**Identify improvement areas**

The purpose of identify improvement area was to recognize the areas with weakness and strengths of the product and analyze which areas needs to be improved. The organizations business goals which were provided needed to be achieved for improvement. The customer expectations must be improved for customer satisfaction. Identify any risk associated with the stated improvement objectives must be achieved. The areas which need improvements were identified by:

* Strengths in assessment process
* Weaknesses in assessment process
* Risks in assessment
* Customer and product owner expectations.
* Additional Risks

The product’s assessment can be improved only if the strengths and weakness of the processes are identified. The processes whose capability level has be achieved according to the defined processes then they are the part of the processes strength. The processes whose capability levels were not achieved included in the weakness process and action must be required to be developed. The risks involved in assessment was necessary to be achieved so that they can handle various risks while assessment. The improvement must be able to identify the customer and product owner’s expectations and must be implemented in action plan.

**Analyze assessments strengths and weaknesses**

The current strengths and weaknesses of the process analysis guided us to identify the process-related risk and indicated opportunities which were required for improvement. The processes which are included in strength and weaknesses process are explained below.

Assessment Strengths

* 7.1.2 Software Requirement Analysis Process
* 7.2.3 Software Quality Assurance Process
* 7.2.4 Software Verification Process

Strong processes were those processes that followed all the best practices while performing rating of each process attribute. These practices could be adapted and instituted in the organization. The software processes (7.1.2) Software requirement, (7.2.3) Software Quality Assurance Process, (7.2.4) Software Verification Process analysis performed all the attributes that were required in the rating of each attribute to reach capability level 2.

Assessment Weaknesses

* 7.1.5 Software Construction Process
* 7.2.6 Software Review Process

The weak processes included those processes which had low process rating attribute and missing practices that were needed to enable process to achieve a process purpose aligned with the specific needs of the organization. The processes (7.1.5) Software Construction Process and (7.2.6) Software Review Process did not achieve the necessary process attributes that were required to achieve the desired capability level. An action plan needs to be developed for these processes to reach level 2 of the capability level.

**Review organizational improvement objectives**

The processes and relationships were analyzed to evaluate the processes that have a direct impact on the organizational development methods. The priority list was then prepared for identifying the processes which needs improvement. The review would guide to recognize which processes need necessary improvement action plan. The organizational improvement were prioritized in the following level

Highest Priority

* 7.1.2 Software Requirement Analysis Process
* 7.2.3 Software Quality Assurance Process

Medium priority

* 7.1.5 Software Construction Process

Low priority

* 7.2.6 Software Review Process

The highest priority processes are those processes which have achieved the all the necessary process attributes to reach the level two of the capability level. These processes have accomplished the fully accomplished objective in process improvement. The medium priorities are those processes which need some improvement to the desired goal. The lowest priorities are those which require more development action plan for improvement.

**Analyze effectiveness measurements**

Analyze effectiveness measurements are those processes which require improvement to reach the specified capability level. Organizations with previous experience in process improvement may already have a measurement. The processes which requires in attaining level 2 of the CMMI model are as follows.

* 7.1.5 Software Construction Process
* 7.2.6 Software Review Process

The software construction process and the software review process have attained level 1 of the CMMI model. An action plan needs to be established in the process performance as both the processes have obtained largely achieved rating and needs to accomplish fully achieved rating for capability level 2. The managed attributes of the software review process needs to perform fully achieved rating as these attributes are still on partially achieved rating and software construction process needs improvement as the rating were largely and partially achieved.

**Derive action plan**

* Improvement Actions:

The necessary defined actions were implemented and process identification for each associated metrics

* Responsibilities:

In this responsibilities are defined as below

1. Program management will be responsible for implementing improvement actions for the Software Construction Process
2. Software analyzing groups will be responsible for implementing improvement actions for the Software Review Process.
3. Technical publications department will be involved with all the phases of the improvement process to ensure traceability elements are maintained throughout the documentation.

* Estimates

1. Cost were estimated by responsible department
2. Advantages of improvement was estimated

* Risks if actions not taken:

1. Customer satisfaction would be achieved
2. Business goals would not be achieve
3. Greater competitiveness would not be achieved
4. There will be investment loss in improvement process

**Step 5 – Implement improvements**

• (Not included in this homework)

**Step 6 – Confirm improvements**

• (Not included in this homework)

**Step 7 – Sustain improvements**

• (Not included in this homework)

**Step 8 – Monitor performance**

• (Not included in this homework)

**Lesson Learned**

We Learned:

We both were studying our first semester as being an international student and this was our first software specific subject, and we gained an enormous amount of knowledge by completing homework assignments. In this assignment we were asked to assess process and realize its capability level. We were asked to perform assessment for a set of process and determine its capability and for the process not achieving capability level 2 we had to develop an action plan which can help the process to achieve capability level 2.

For completing this homework, we had to learn ISO 12207 which is software lifecycle process. It is divided into two main process group namely System context processes and software specific processes. It consists of seven process groups, in total these 7 process groups consist of 43 different processes. Also we were asked to learn ISO 15504 which is Software engineering process assessment(SPICE). ISO/IEC 15504 Process Assessment (SPICE Model) is a model to guide how to perform process assessment in terms of process capability. It consists of 10 parts and can vary in future. It identifies the capability of different processes and requirements for performing an assessment.

Using the two ISO standards we had to perform mapping and gapping of our scrum and xp practices performed in our assignment 1. So we learned the mapping of scrum and xp to ISO 12207. Also after that we learned finding the gaps between scrum, xp practices with ISO 12207. After performing the mapping and gapping we started rating the process with the help of outcomes and ISO 15504. Which help us understand the capability level of the process which is assessed. Also ISO 15504 help with developing an action plan for some specific process which has not achieved desired capability level.

We plan to learn this process in more detail in near future. Getting to work on project assessment in real industry can help us gain more and thorough knowledge about these methodologies. Also we will like to learn similar methodologies like these in future.

Chirag Padsala:

Being my first software specific subject as an international student, it was completely new to me. I learned Scrum and XP for the first time and successfully implemented it for developing an android application. After completing the application development, I had to do learn ISO 12207 for software life cycle process and ISO 15504 for process assessment which was kind of confusing at first. But after creating a plan and learning each process in detail and in conjunction. After learning the process, I gained knowledge about ISO 12207 which is software life cycle process. It is divided into two main process group namely System context processes and software specific processes. It consists of seven process groups namely Agreement Processes, Organizational Project-Enabling Processes, Project Processes, Technical Processes, Software Implementation Processes, Software Support Processes, Software Reuse Processes. In total these 7 process groups consist of 43 different processes. ISO/IEC 12207 was published on August 1995, and was the 1st International Standard to provide a set of lifecycle processes, activities and tasks for software development.

After that I learned ISO 15504 which is Software engineering process assessment(SPICE). ISO/IEC 15504 Process Assessment (SPICE Model) is a model to guide how to perform process assessment in terms of process capability. It consists of 10 parts and can vary in future. It identifies the capability of different processes and requirements for performing an assessment, Process Reference Models, Process Assessment Models, Verifying conformity of process assessment. Process assessment is based on 2-dimensional model 1. A process dimension and 2. A capability dimension.

As a part of my project I learned this two ISO model for assessing our project which was creating an insertion sort app using scrum and xp practices. I thoroughly learned how the software lifecycle can be used for mapping agile processes and finding gaps using this model. Once the mapping is done and gaps are identified rating of process is done which help us know to capability level of the process. As per our assignment we were asked to assess specific process and realize the capability level of those process, if the process achieves capability level 2 we needed to stop and if it doesn’t we had to develop an action plan which will depict how the process can achieve capability level 2.

I would like to learn more about process assessment in near future by getting into a well-known organization. Getting a job for process assessment can also help me get the real world scenario about process assessment of a particular project.

Prathmesh Pardhiye:

As per our first homework assignment I learned how to develop a project using scrum and XP practices. As scrum and XP was new to me it was a bit tough implementing it at first but after learning it thoroughly I got a brief idea about these agile methodologies. After successfully completing the first assignment we were asked to perform process assessment for the project developed in homework one using different ISO standards.

I started learning the ISO 12207 and ISO 15504 for gaining an overview about the software lifecycle process and software process improvement and capability. We performed the mapping and gapping by comparing to ISO 12207. I learned how to mapping and gapping is actually done and followed by that how the rating is done to assign capability level to a specific process. And also I learned how to develop an action plan for the gaps identified during rating process. Which can help the process to achieve desired capability level.

I would like to get a job in a good and well-known organization for performing software process assessment, which will give me a practical and real world challenges to deal with.