Seminar 1:

Technical product management:

Product management is one of the important organisation roles. It is the process of collecting and using data on the products that a business or organization sells handles or makes. Product management process can be handled by the project managers in companies. Some product managers have natural affinity for working with development, some for sales and marketing and some others prefer for work on business issues. Among these, technology expertise plays an important role and deals about how the product works. The product managers pick up a deep understanding of the product and its technological capabilities. And also market-driven environment generates large amounts of requirements from multiple sources, internal and external, threatening to overload the technical management of products [1]. The term technical project management describes a person, not a role. It implies that it does not describe a product manager who needs to actually perform technical tasks (software architecting and coding), but they need to perform product management role in close coordination with the technical team. Technical product managers should have strong technical background and this position generally involves close interaction with development leads, product architects and key customers. The activities to be performed by a technical manager include:

1. Conducting technology assessments.
2. Analysing the competitive overlook.
3. Maintaining the product portfolio landscape.
4. Monitoring and organising industry innovations.
5. Writing product requirements and use case scenarios.
6. Maintaining a status dash board for all the portfolio products.

Seminar 4:

What tools are available for requirements management?

Requirements management: Requirements management is the process of collecting, analyzing, refining, and prioritizing product requirements and then planning for their delivery. The main purpose of requirements management is that it helps to ensure that the organization is meeting the needs of customers and stakeholders.

Tools:

A Requirement Management Tool (RMT) has specific database capabilities that are needed for effective management of the requirements [2]. There are several tools available for requirements management and some of the important tools are:

- IBM Rational Doors
- Jira
- Jama
- Kovair ALM Studio
- ALM studio

- Mingle
- Caliber
- inteGREAT
- in-STEP BLUE
- TestTrack RM
- Blueprint
- Enterprise Architect

Any particular tools for agile requirements management?

There is no comprehensive definition of agile requirements management. It is an evolving collection of practices. Although agile has made many important contributions to RM, mature and comprehensive agile RM approach is still lacking. Every project or organization should adopt its own suitable approach to agile RM.

Some of tools available for agile requirements management are:

- Agile Manager
- CA Agile Central
- Jira
- Mingle
- VersionOne
- Rally
- AgileZen
- Agilo
- Kanban tool

Seminar 2:

How to connect requirements to the architecture?

Writing clear, testable and atomic requirements that speak to an actual need rather than design in disguise is a challenging thing to do. Hence requirements are often captured as parent-child relationships, where a given child requirement is supposed to be refined statement relative to the parent. In general, requirements are connected to the architecture by using use case diagrams. These diagrams help the designers and architect to easily understand and form the design and architecture of the system. In recent years, software engineers are facing a big challenge when conceptualizing and designing large-scale software applications. And hence they are in a need of technique which produces an architectural-level view of the software design directly from its requirements that satisfies both functional requirements and design quality attributes [3].The UML is introduced as a new way to describe functional requirements, the Use Case. Use Case diagrams describe how a user of the proposed system will interact with this system to perform a discrete unit of work. Each diagram describes a single interaction over time that has meaning for the end user. Security patterns are also a promising tool to transition from requirements to architecture. In particular, pattern descriptions are expected to contain sufficient contextual information to assist an architect with selecting a suitable pattern for a particular problem. This is not always the case with security patterns, however.

Summary of article (v.d. Weerd & Brinkkemper "Towards a reference framework for software product management"):

The authors of this paper presented a reference framework for software product management which mainly identifies key process areas such as portfolio management, product road mapping, and release planning and requirements management. Software product management has received much practical attention in the recent years. Hence the main goal of this paper is to provide a structure for the body of knowledge of software product management through a reference framework which is based on an overview of state-of-the-art literature and industrial case studies which is beneficial for research and practice in many fields. This framework provides starting point for definition of key terms and identification of open RQ's and development of improved and integrated tool support. The basic structure of this framework is mainly based on the objects or artefacts of product management and upon the set of stakeholders identified in the scope of work of the product manager. As already specified, the reference framework comprises of four major parts.

1. Portfolio management: It mainly helps to encompass the decision making related to the set of existing products, introducing new products by observing the market trends and also the product development strategy. It is viewed on the top of the reference framework. The input for this process is given by the board, market and partner companies which help to take important decisions about the development strategy and lifecycle. Through this, a software product line is identified and it serves as an input for product road mapping. It involves four tasks

- Partnering and contracting
- Market trend identification
- Product life cycle Management and
- Product line identification

2. Product road mapping: It is used to identify different releases of a product. It is complex due to dependencies on other related products, technology changes and the distributed development. This process receives input as product lines from portfolio management and it helps in identifying themes and core assets. The collected information is specified in the road map which further acts as an input to the requirements organisation. . It involves mainly three tasks

- Theme identification
- Core asset identification and
- Roadmap construction.

3. Requirements management: It contains the activities such as gathering, identifying, revising and organizing incoming requirements from various stakeholders. And it starts with gathering all requirements from both internal (within the company) and external stakeholders Input for this process is obtained from different stakeholders. All the product requirements are grouped according to the product and core asset. And it includes three tasks

- Requirements gathering
- Requirements identification and
- Requirements organizing

4. Product release planning: It mainly deals with the set of requirements that are to be released. The input for this is obtained from the requirements management process where the obtained list of requirements is prioritized. A release definition is prepared for the selected requirements and validated by different stakeholders (internal) and a business case is sent to the board of approval. Once the approval is received, launch preparation package is delivered to the stakeholders. . It involves five tasks

- Requirements prioritization
- Requirements selection
- Release definition
- Release verification and
- Launch application

Hence the framework proposed in this article serves as a first step to place the software product management domain in the field of scientific research in requirements engineering and also helps to provide a structure for an integrated body of knowledge.

D. Leffingwell "Scaled Agile Framework":

This website gives a clear description of lean software and systems engineering and scaled agile framework. SAFe is an online, freely revealed knowledge base of proven success patterns for implementing Lean-Agile software and systems development at enterprise scale and also provides guidance for work at enterprise portfolio, value stream, program and team levels. In SAFe 4.0, four levels are described and all the important topics in each level are also described clearly. This blog includes

1. Portfolio: This level provides guidance for strategy formulation and portfolio communication, organising and funding value streams, managing the flow of larger initiatives and governance.

Some of the important concepts in this level include:

a) Program portfolio management (PPM): It represents the people who have highest-level strategy and decision-making responsibility in the framework and who have the primary responsibility for strategy and Investment Funding, Program Management and Governance within a specific SAFe portfolio. PPM has the responsibility to participate in the establishment and communication of strategic themes that guide the enterprises investments and strategy. And also during the transition to Lean-Agile, the PPM function may need to fulfill its responsibilities for both Agile and traditional waterfall programs.

b) Epic owners: Epic owner is responsible for driving individual epics from identification through analysis of the portfolio Kanban system and decision making process of Program Portfolio Management. And this role has the responsibility of preparing the epic, presenting it and implementing the epic after its approval. And this role also helps to fill in the gaps that often occur in organizations with high-level initiatives descend from the top of the organisation to implementation.

c) Enterprise architect: Enterprise Architect works with business stakeholders and solution and system architects to drive technology implementation across value streams. The primary responsibilities of enterprise architect include maintaining high level, holistic vision of enterprise solutions and development initiatives, help to define key technical initiatives that support budget via enabler epics.

d) Portfolio Backlog: It is the highest-level backlog in SAFe and it provides a holding mechanism for the upcoming business and enabler epics required to create a holistic portfolio solution set. It holds and prioritizes epics that have been approved for implementation.

e) Portfolio Kanban: This system is used primarily to address the flow of epics that affect the course of action for value streams and agile release trains that realize them. And it describes a number of stages that an epic passes through on the way to implementation and collaboration that is required for each stage.

2. Value stream: This level offers guidance for companies building the world’s largest software systems which include Solution Intent, solution management, engineering and architecture, agile customer and supplier relationships and value stream coordination.

Some of the important concepts in this level include:

a) Solution Intent: It is a critical knowledge repository to store, manage and communicate what system builders are building and how they are going to build it.

b) Product and Solution Management: Solution management and product management are the main content authorities in SAFe which helps to guide the value stream and program levels.

c) Release Management: It assists with planning, managing and governing releases of the solution and has the authority and responsibility to help guide the value stream toward the business goals.

d) Value stream epics: Epics are the containers for significant initiatives that help to guide value streams towards the larger aim of the portfolio.

e) Value stream Kanban: Helps to facilitate the flow of value, solution builders and apply a Kanban system at the program and value stream levels. These help teams to increase visibility into new work and the flow of work.

f) Value stream Backlog: These are repositories for all the upcoming work that affects the behaviour of the solution.

3. Program: This level describes agile release trains, teams of agile teams that build solution capabilities and subsystems. And the agile release trains align teams to a common mission, provide architectural and user experience governance, facilitate flow and provide objective evidence of progress and fitness for purpose.

The main concepts in this level include:

a) System Architect/Eng: It aligns the value stream and agile release trains to a common technological and architectural vision of the solution under development.

b) Program epics: These epics are initially captured in the portfolio Kanban and move through this system under work-in-process limits and it helps to assure that those doing the work has the time necessary to conduct responsible analysis.

c) Program Kanban: It helps to facilitate the flow od value, solution builders apply a Kanban system at program and value stream levels.

d) Program Backlog: The program and value stream backlogs are the repositories for all the upcoming work that affects the behaviour of the solution.

4. Team: SAFe teams are self-organising, self-managing and cross-functional teams and these teams apply Scrum, XP and Kanban and deliver valuable, tested and working software and systems every two weeks.

The main concepts of this level include:

a) Agile team: It consists of a small group of dedicated individuals who work together and have skills necessary to define, build and test increments of value in a short time box.

b) Product owner: Product owner is the member of the Agile Team who serves as the Customer proxy and is responsible for working with product management and other stakeholders.

c) Scrum master: It is a special role for an Agile Team member who spends much of his/her time helping other team members to communicate, coordinate and cooperate and assists the team in meeting their delivery goals.

d) ScrumXP: It is a self-organising, self managing and cross-functional group of five to nine people collocated wherever possible.

e) Team Kanban: It helps teams to facilitate the flow of value by visualizing work flow, establishing work-in-process limits, measuring throughput and continuously improving their process.

Seminar 3:

Wnuk et al. "What Happened to Our Features? Visualization and Understanding of Scope Change Dynamics in a Large-Scale Industrial Setting":

The authors of this paper argue that in the case of the development of software platforms for product lines, the decisions on which features to implements are affected by changing markets and evolving technologies. Hence it is difficult to decide which requirements to be included in the scope of the upcoming project. The process of selecting of subset of requirements to a forthcoming project and postponement of the implementation of other requirements is known as scoping and is considered as an important activity in order to achieve economic benefits in product line development. Hence the problem of handling the scoping process is addressed by the authors of this paper by proposing the concept of Feature Survival Chart (FSC) and is acknowledged by the practitioners as a valuable support for the selection of requirements. For this, this article extends the contributions of past research by deriving findings from industrial application in three projects (by conducting a case study) and scope tracking measurements. The proposed technique helps to recognize the problem of setting large scope against the available resources and also increases understanding of the consequences of setting a limited scope early. The feature survival graphs helps us to identify which features and which time frames to analyze in order to find scoping issues related to uncertainties in the estimations on which the decisions rely. The case company is a large company with more than 5000 employees which develops embedded systems for a global market and is using a product line approach. It handles the requirements as primary flow and secondary flow and is associated with several groups of specialists in various stages of requirements management process. The company uses a stage-gate model with several increments and there are milestones and tollgates to control the progress of project. The development of the FSC chart and corresponding scope tracking measures was performed in an interactive manner that involved practitioners from the case company. And to gather data for this study, authors have implemented an exporter to retrieve the data from the scope parameter of each feature in the Feature List document. After creating graphs, a meeting with practitioners was held in order to present and discuss results as well as address issues for future work.

The Feature Survival Chart is designed in such a way that the scope changes are clearly viewed over the time and is illustrated on X-axis and each feature is placed on a specific place on the Y-axis. And the various scope changes are visualized using different colours and hence the scope changes can be viewed as a change of the colour. The chart was obtained through the case study conducted in the company and corresponding conclusions are drawn from the chart based on the colour changes with respect to the milestones. Further, this research is extended by formulating questions related to external attributes, which in turn related to external and a set of five measurements (time related measurements and feature related measurements). This idea is driven by the fact that measurement is an effective mechanism for characterizing, evaluating, predicting and providing motivation for various aspects of software construction processes. All the measurements are clearly described and theoretical analysis of these measurements is also presented in this paper. Time-related measurements are presented as a function of time, while feature-related measurements are

presented in the form of distributions for each evaluated project. And these measurements are empirically applied on the projects and corresponding results are presented.
Hence the visualization technique presented in this paper provides feedback about ongoing scoping activities and also helps for the visualization of past project scoping activities. And the measurements presented in this paper complement the visualization technique by quantitative characterization and qualitative rationale for scoping decisions. And it also helps to give a better overview of the scoping process of the whole project on a single page size graph. From this article, it was concluded that this method can be applied to large scale projects, and thus it demonstrates the scalability of the method.

Reflections on contribution of papers with respect to seminar discussion questions:

All the articles given in all the seminars are very useful to obtain answers for the questions that are discussed in seminar and also provided useful information relating to many important concepts in the case of large scale requirements engineering and market-driven requirements engineering. The contributions of the articles with respect to the questions are discussed below.

Seminar 0:
Four articles were given in this seminar and one of them ("The art and science of release planning") is related to release planning and involves description and application of two ways of performing release planning. And hence through this article, I have got an idea about what is release planning and how it can be applied practically and some of the challenges faced during the release planning. So this information is helpful for me to answer few questions in seminar 0 and seminar 1 that are related to release planning and market-driven requirements engineering (such as what is a release?, how do we do release planning etc). And two of the articles ("A case study evaluation of the guideline-supported QUPER model for elicitation of quality requirements" and Introducing support for release planning of quality requirements – an industrial evaluation of the QUPER model") are related to the quality requirements and it helped me to know how the non-functional requirements are handled in the companies and the importance of their consideration in practical situations. These two articles helped me to answer some of the questions in seminar 0, seminar 1 and seminar 4 that are related to the release planning and quality requirements (Ex: How do you connect quality requirements into the mix? How do quality requirements impact other requirements?). And the article ("A market-driven requirements engineering process: results from an industrial process improvement programme") helped me to gain important information about how large scale requirements are handled practically and about market-driven requirements engineering. And this article helped me to answer questions related to large scale requirements engineering and market-driven requirements engineering and continuous flow of requirements. This article also introduced the concept of milestones and helped to answer questions like (how shall we decide to which milestone an issue belongs?).

Seminar 1:
Five articles were given in this seminar and article ("Requirements Abstraction Model") helped me to gain a lot of knowledge about how continuously changing requirements and how the initial set of requirements arrive (in different abstraction levels) and how the companies handles them and manage the complexities in a systematic manner. And this article helped me to answer questions that are related to grouping of requirements,

continuously changing requirements and its challenges. Article ("A method for early requirements triage and selection utilizing product strategies") also deals with how to handle the continuous flow of requirements in practical situations and how to handle large set requirements coming from different stakeholders and many sources and this article helped me to answer the questions related to continuous flow of requirements and also changes in the requirements over time. Article ("Requirements Engineering. In search of dependent variables") mainly focuses on how the changes made in requirements engineering processes impact the other dependent variables and this article helped me to answer the questions that are related to the dependencies and how the changes impact the company as well as the product. Article ("Quality Requirements in Industrial Practice– an extended interview study at eleven companies") mainly deals with the importance of quality requirements and challenges that are faced by the companies by not considering these requirements properly and hence this article helped me to answer the questions related to quality requirements (How do you deal with differing quality requirements in a product line setting?). And the article "A cost-value approach for prioritizing requirements" deals with the prioritization of requirements based on their cost and value. This article helped me to answer the questions related to prioritization of requirements such as (How shall we prioritise among the requirements?).

Seminar 2:

Six articles were given in this seminar and all the articles provide important information regarding software product management, market-driven requirements engineering for software products, issues related to over scoping and how the decisions affect the outcomes of the project. Article ("Towards a reference framework for software product management") provides a framework through which the software product management concept is clearly understood and this article helped me to answer the questions related to software product management and also questions related to release planning and road mapping. Article ("Market-Driven Requirements Engineering for Software Products") helps to provide valuable information regarding market-driven requirements engineering related to the software products and helps to answer the questions regarding challenges faced by market-driven companies and software products. Website "Scaled Agile Framework" helped me to understand new concepts such as epics, Kanban, Backlog etc and also few important concepts such as road mapping, releases and different roles of scrum. This article helped me to answer the questions related to these topics (like how do you transform epics into reasonable chunks of work?). Article ("Are You Biting Off More Than You Can Chew? A Case Study on Causes and Effects of Over scoping in Large-Scale Software Engineering") mainly focuses on the problem of over scoping, corresponding challenges and how this can be handled by the companies practically and helps to answer the questions related to scoping. Article ("Factors Affecting Decision Outcome and Lead-time in Large-Scale Requirements Engineering") mainly deals with the continuous changes of requirements and how decisions affect the outcomes of the project and helped me to answer the questions related to changes in requirements flow. Article ("An industrial survey of requirements interdependencies in software product release planning") mainly concentrates on the interdependencies between different requirements are handled in different companies and to get in-depth knowledge about these dependencies. And thus it helps to answer the questions related to requirements interdependencies such as (Requirements are dependent on each other! How to deal with this?).

Seminar 3:

Two articles are given in this seminar and article ("What Happened to Our Features? Visualization and Understanding of Scope Change Dynamics in a Large-Scale Industrial Setting") deals with the changes in the scope of the project and how these changes can be recognized and handled in future practically. This paper helped me to answer the questions related to scope changes of the project such as (What scope changes are likely to happen?).
Article ("Obsolete Software Requirements") deals with the inclusion and exclusion of obsolete software requirements in the scope of the project and also explains challenges that may arise due to such requirements. And this article helped me to answer the questions related to the scope of the project.

Seminar 4:
The article mainly concentrates on the challenges faced by the market-driven companies due to continuous integration and this helped me to answer the questions related to the challenges faced in the case of large scale requirements engineering.

Hence all the given articles provided useful information relating to the questions discussed in the seminar. However, I have found that some of the questions are not answered by the seminars completely and to answer those questions, I have referred to other related articles and websites related to large scale requirements engineering.

References:

[1] Gorschek, T. (2006). *Requirements Engineering Supporting Technical Product Management. Technology.*

[2] T. Hammer and L. Huffman, "Automated requirements management-beware HOW you use tools: an experience report," *Requirements Engineering, 1998. Proceedings. 1998 Third International Conference on,* Colorado Springs, CO, 1998, pp. 34-40.

[3] Schmidt, Herjens, J. (2011). Connecting security requirements analysis and secure design using patterns and UMLsec.