ENSE701
Contemporary Issues in Software Engineering

Week 5 Lecture : Requirements Prioritization in Scaled Agile Distributed Software Development

What is requirement?

- It is about What not How
- It is about need
- IEEE – A condition or capability needed by user to solve a problem or achieve an objective.

What is Requirements Engineering (RE)

- RE, which is a branch of Software Engineering (SE), deals with discovering, specifying, analysing, and documenting the requirements of a system.
- Each software development process goes through the phase of requirements engineering

Requirements engineering processes

- The processes used for RE vary widely depending on the application domain, the people involved and the organisation developing the requirements
- However, there are a number of generic activities common to all processes

  Requirements elicitation, analysis and prioritization

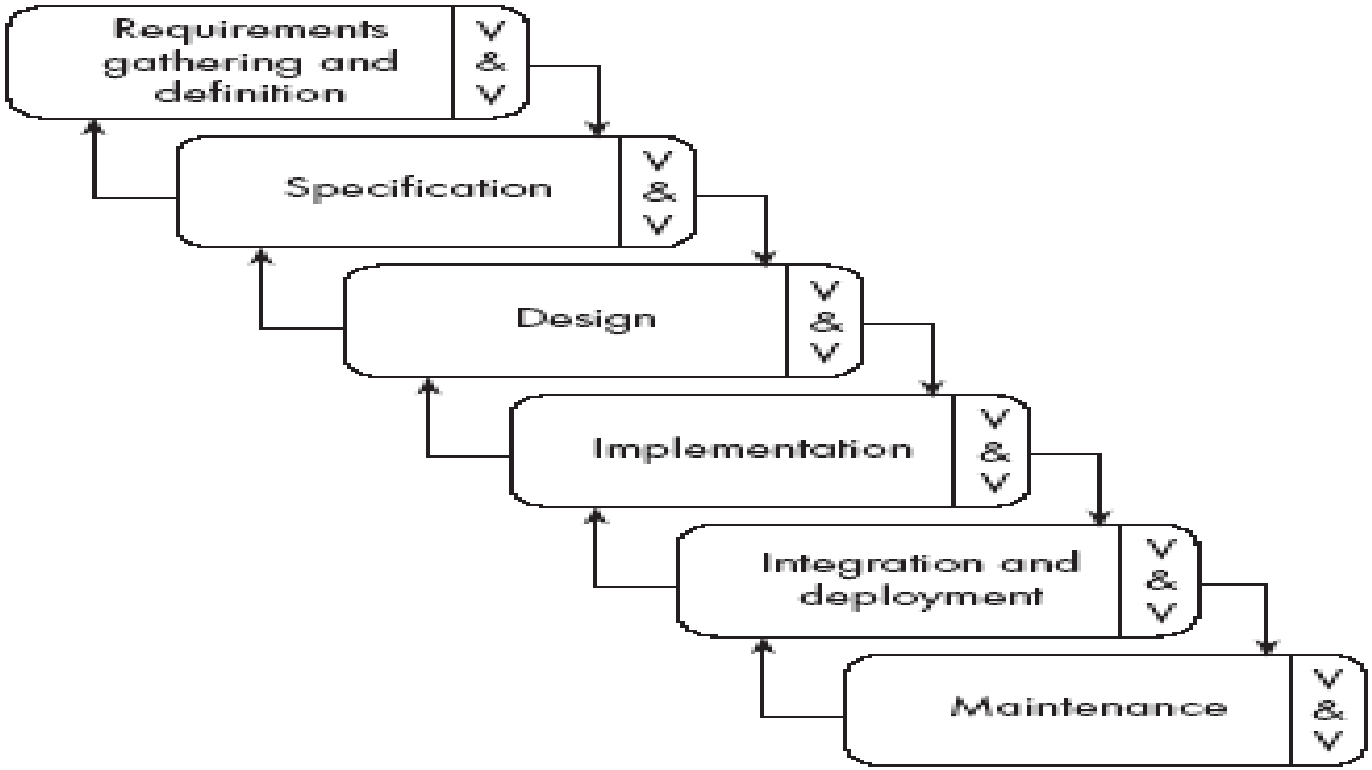
  Requirements specification

  Requirements validation

  Requirements management

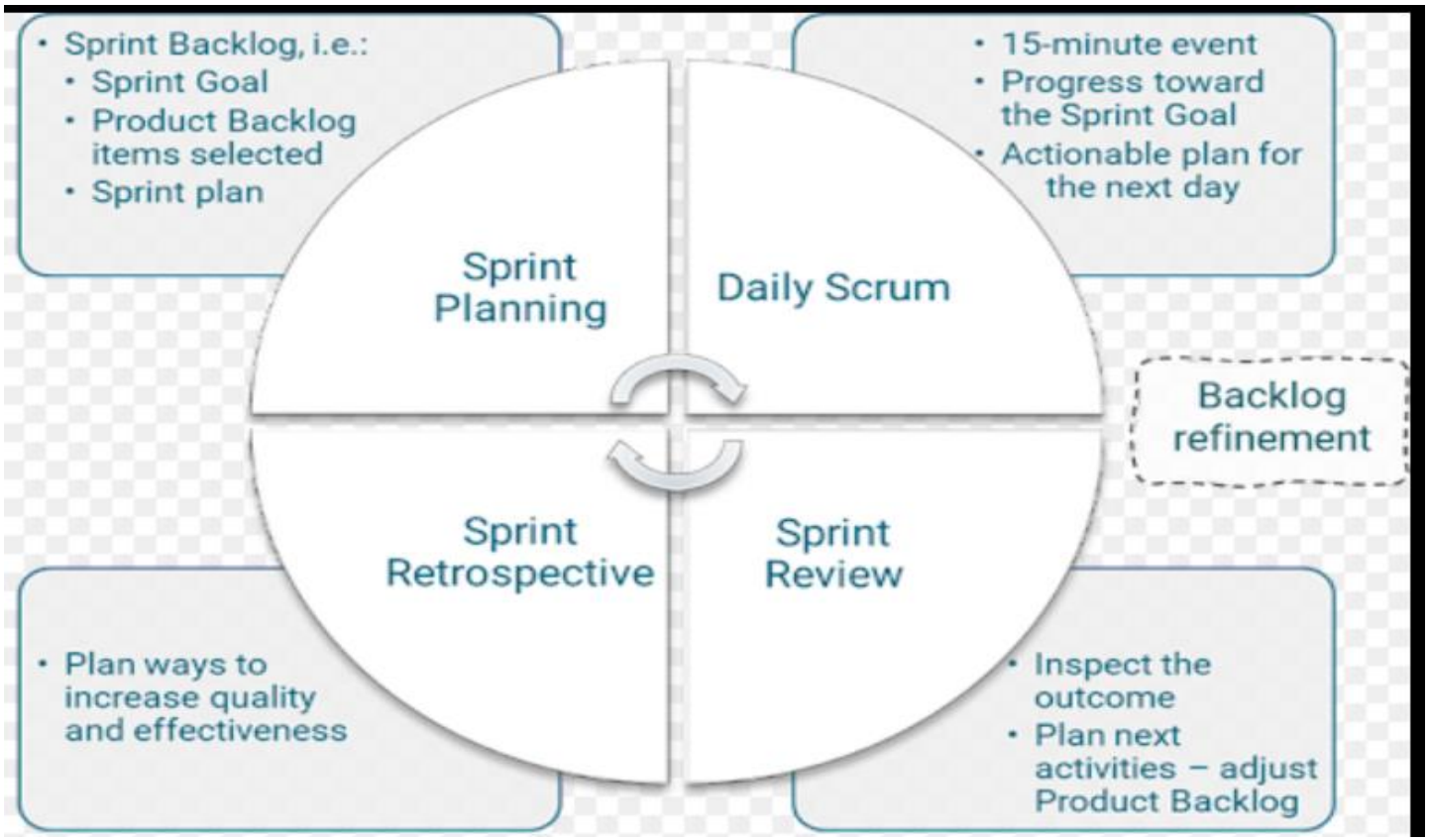
RE and software development methods

- RE in classic development methods (e.g., Waterfall)



RE and Software development processes

RE in Agile development (e.g., Scrum)



RE and Software development processes

- RE in Agile development (e.g., Scrum)

RE activity	Scrum implementation
Requirements Elicitation	• Product Owner formulates the Product Backlog. • Any stakeholders can participate in the Product Backlog.
Requirements Analysis	• Backlog Refinement Meeting. • Product Owner prioritizes the Product Backlog. • Product Owner analyzes the feasibility of requirements.
Requirements Documentation	• Face-to-face communication.
Requirements Validation	• Review meetings.
Requirements Management	• Sprint Planning Meeting. • Items in Product Backlog for tracking. • Change requirements are added/deleted to/from Product Backlog.

Main topic of today's lecture: Requirements prioritization in Scaled Agile Distributed Development

- Adoption of basic Agile methods at the enterprise level
- Scaling Agile frameworks (e.g., DAD, SAFe) to support enterprise-wide agility and scale agile practices

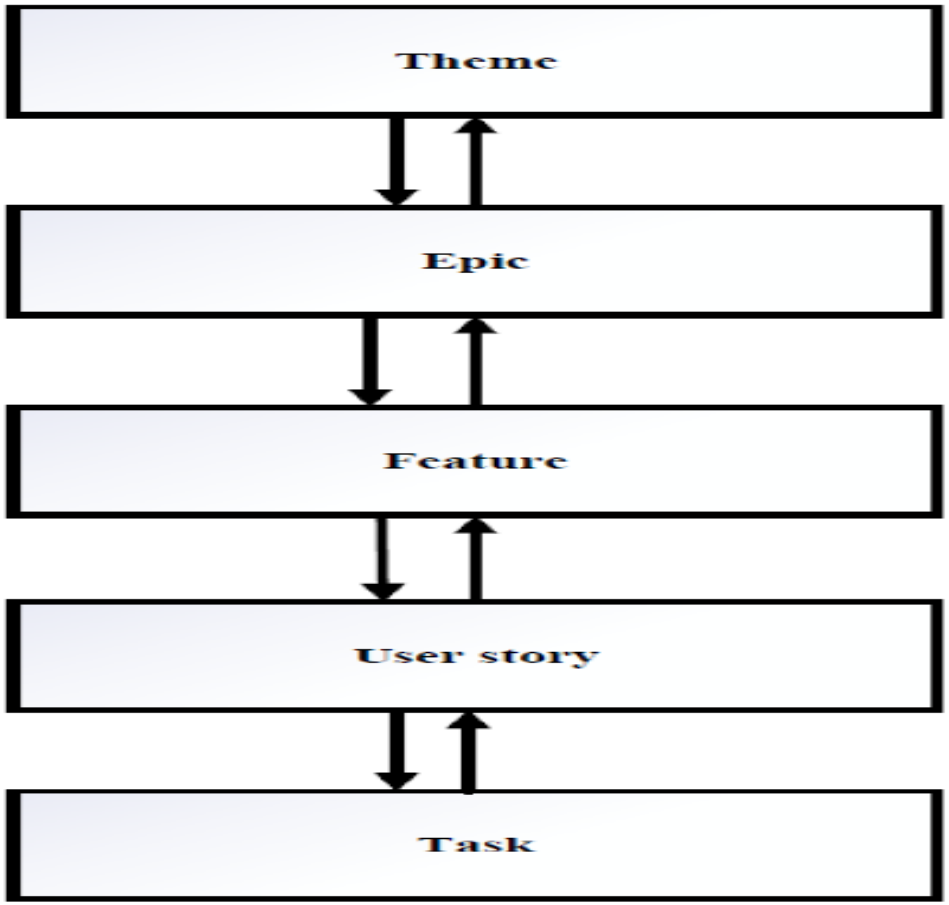
Requirements prioritization in Scaled Agile Distributed Development

- Prioritization of requirements as decision making process
- Prioritization is beyond the team level in scaled Agile development

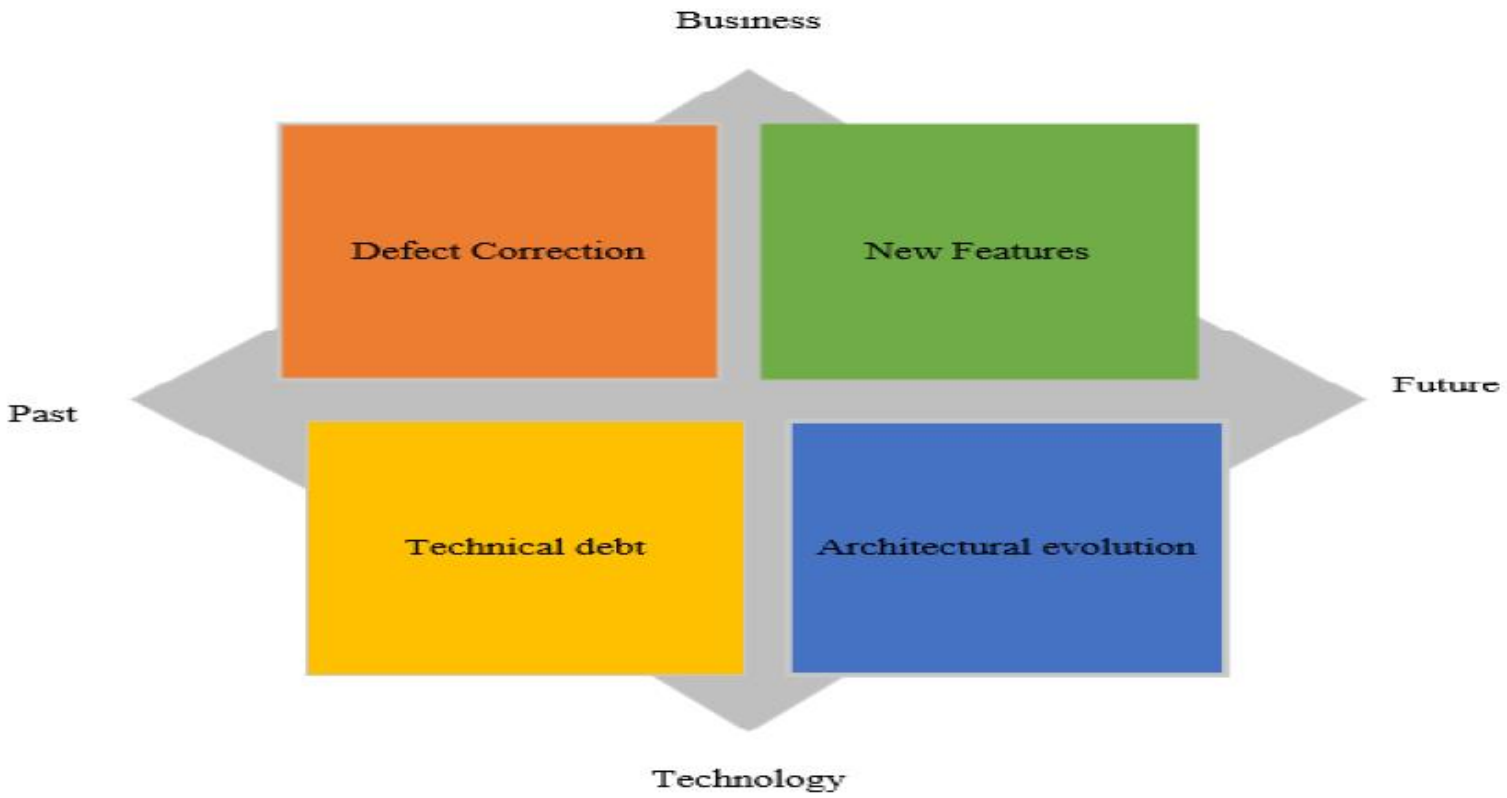
Key decision-making levels

- Portfolio level
- Domain level/Program level
- Team level

Flow of requirements across scaling agile decision-making levels

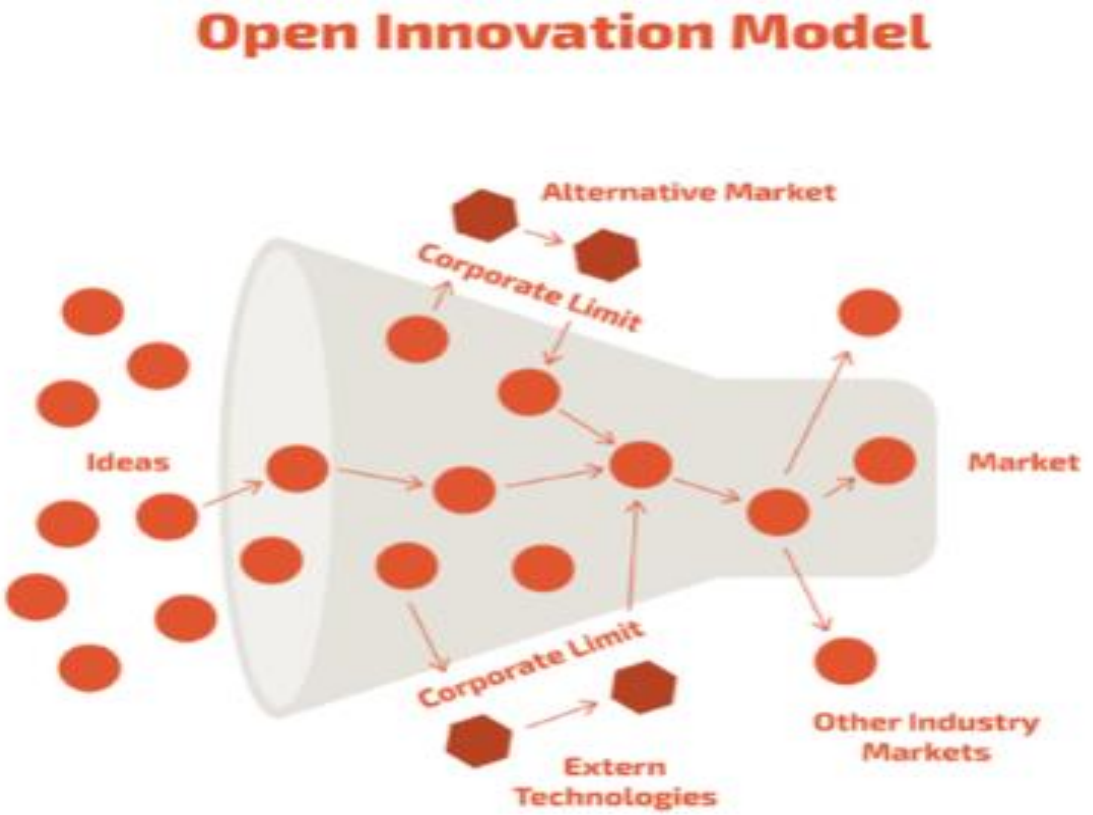


Requirements classification across scaling agile decision-making levels



Open innovation approach for requirements discovery across scaling agile decision-making levels

- Internal sources (e.g., portfolio management, product management, developers)
- External sources (e.g., customers, industry analysts, market research)



Decision-making at Portfolio level

Two main objectives at the portfolio level in terms of decision making:

- Business goals that connect with organization's overall business strategy are decided and prioritized
- HLRs towards achieving business goals are formally signed off, communicated, and allocated to the engineering for implementation

Key practices for Decision-making at Portfolio level

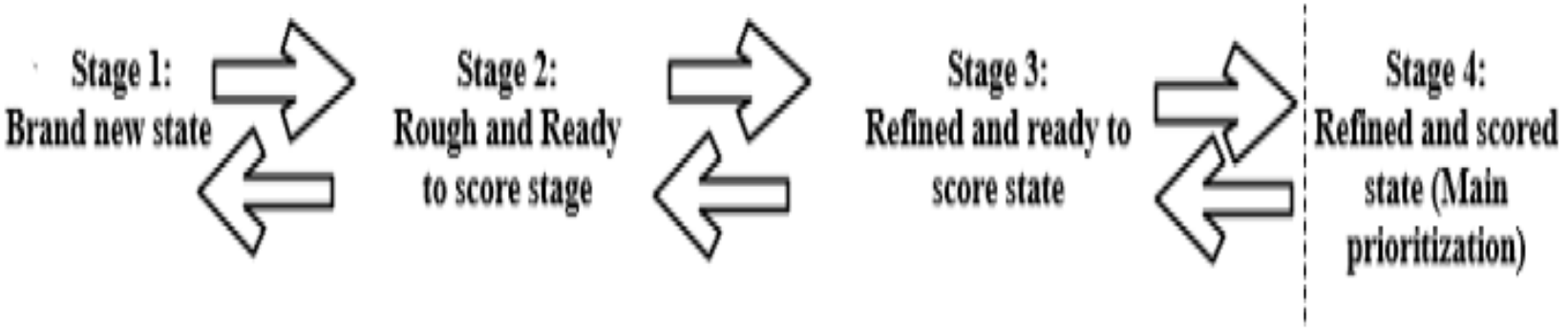
- Portfolio level – supported via inter-iteration prioritization
- Portfolio management – cross-functional decision-making team
- Continuous decision-making – typically quarterly, monthly/fortnightly check-ins, ad-hoc meeting if needed
- Combination of quantitative as well as qualitative practices

Domain level/Program level

Plays two-fold role during decision-making

First part of domain level:

- Initial decision-making on HLRs that emerge from each of the domains towards achieving strategic themes and/or contributes to the formation of strategic themes.
- Typically supported via intra-iteration prioritization


Stage 1:
Brand new state
Stage 2:
Rough and Ready
to score stage
Stage 3:
Refined and ready to
score state
Stage 4:
Refined and scored
state (Main
prioritization)

Domain level/Program level

Second part of domain level:

- Inter-iteration prioritization across teams (project specific)
- Decomposition of HLRs to implement them via short development cycles

Team Level

- Typically form intra-iteration prioritization
- Generally employed basic Agile methods (e.g., Scrum) for decision-making on the priority of requirements
- Detailed requirements provide required information to the delivery teams (e.g., user story)
- Priority decisions within the team

Boundary spanning mechanisms across scaling agile decision-making levels

- Boundary spanning is the act of bringing together two or more groups of people who are typically separated by location, that is, locally distributed and/or globally distributed., or separated by hierarchy, or a function

- Boundary spanning yields knowledge acquisition, negotiation, consensus building, and conflict resolution which are the key activities typically needed while making decision on the priority of requirements

Boundary spanning mechanisms

Three categories of boundary spanning mechanisms

- Boundary spanning event(s),
- Boundary spanning requirement artefact(s),
- Boundary spanner role(s)

Collaborative technologies (CTs) across scaling agile decision-making levels

- CTs enable collaboration with distributed members of a decision-making team
- Enable shared understanding of priority of requirements across scaling agile decision-making levels
- CTs- as synchronous and asynchronous mechanisms for decision-making across scaling agile levels
- Spreadsheet, Jira, AHA, ADO, Confluence, PowerPoint, MSTeams, Email

Usefulness of CTs during decision-making

CTs	Role of CTs for requirements prioritization across scaling agile levels				
	Requirements discovery and understanding - elicitation of requirements - knowledge sharing- shared understanding of requirements	**Requirements artefacts**	**Decision making practices- priority score matrix**	**Decision making events: collaborate with cross site members over requirements**	**Scaling agile levels**
Spreadsheet	For knowledge sharing- *shared understanding of requirements*	Spreadsheet as an informal Portfolio backlog		decision making event(s) performed at the portfolio level	Primarily employed at the Portfolio level, Can be used at the other levels as well (i.e., domain level, team level)
PowerPoint	For knowledge sharing- *shared understanding of requirements*			decision making event(s) performed at the portfolio level	Primarily employed at the Portfolio level, Can be used at the other levels as well (i.e., domain level, team level)
AHA	Requirements elicitation- *Online customer portal configured via AHA* Knowledge sharing- *shared understanding of requirements*	Backlog (i.e., Portfolio backlog), template configured for HLRs, template configured for strategic themes,	Automated scoring matrix for HLRs configured in AHA	decision making event(s) performed at the portfolio level, 1st part of domain level	Portfolio level, Ist part of domain level

Usefulness of CTs during decision-making

CTs	Role of CTs for requirements prioritization across scaling agile levels				
	Requirements discovery and understanding - elicitation of requirements - knowledge sharing- shared understanding of requirements	**Requirements artefacts**	**Decision making practices- priority score matrix**	**Decision making events: collaborate with cross site members over requirements**	**Scaling agile levels**
MSTeams Features of MSTeams *Screen sharing, Instant messaging, Team specific channel, Tag to notify people, recording of decision-making events, Offline chat audio/video call, web conference*	Elicitation of requirements, knowledge sharing- *shared understanding of requirements*			decision making events at all the levels for synchronous and asynchronous communication	Portfolio level, Domain level, Team level
E-mail	For knowledge sharing- *shared understanding of requirements*				Least commonly used asynchronous CTs. Employed at Portfolio level, domain level, team level

Conceptual framework of requirements prioritization in SADSD