

Requirements Engineering

Donald G. Firesmith, Firesmith Consulting, U.S.A.

Abstract

In my first column, I introduced the OPEN Process Framework (OPF), which provides an industry-standard terminology for organizing and communicating process engineering concepts. In it, I also introduced the reusable requirements engineering work products that are part of the OPF. This second article introduces the remaining reusable process components that are useful for requirements engineering: specifically, the requirements work units, the producers of the requirements work products that perform these work units, and the associated languages that are involved in requirements engineering.

1 REQUIREMENTS WORK UNITS

Requirements engineering is an activity, one of several kinds of work units that are performed on endeavors. As the following simplified figure illustrates, work units can be categorized (in order of decreasing scope) activities, tasks, and techniques, whereby:

- An **activity** is the highest-level work unit consisting of a cohesive collection of one or more tasks that are performed by one or more collaborating producers when producing a set of one or more related work products or providing one or more related services.
- A **task** is a mid-level work unit that models a functionally cohesive operation performed by one or more producers.
- A **technique** is a low-level work unit that is the reified implementation of (i.e., way of performing) one or more tasks.

Requirements Engineering Activity

Requirements engineering is the activity consisting of the cohesive collection of all tasks that are primarily related to an endeavor's requirements. The typical goals of requirements engineering are to:

- Determine what:
 - The business, application, framework, component, or center must **do**.
 - It must **know**.
 - **Qualities** it must have.

- **Interfaces** it must have to external systems.
- **Constraints** it must conform to.
- Produce and maintain a complete set of high-quality requirements including:
 - Functional Requirements
 - Data Requirements
 - Quality Requirements
 - External API Requirements
 - Constraints
- Produce requirements work products that are required as inputs by other project teams (e.g., the architecture and independent testing teams).
- Thereby help ensure that all project effort is based on the implementation of properly approved requirements.
- Minimize the risk of endeavor failure due to poorly engineered requirements.
- Reduce endeavor costs related to requirements.
- Improve the quality of all endeavor deliverables.
- Increase customer and user satisfaction.

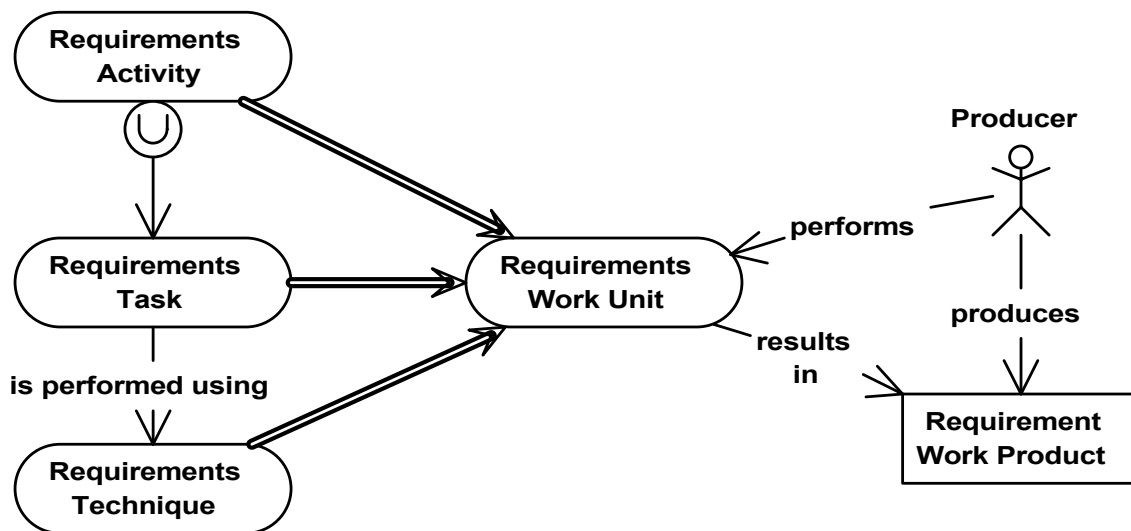


Fig. 1: Requirements Work Units

To achieve the above goals, the objectives of requirements engineering are to:

- Produce a formally documented consensus among all project stakeholders (e.g., customer representatives, users, project champions, project management, developers) concerning either the:
 - Vision of the [re]engineered business or the next version [incremental iteration] of the application.
 - Requirements for the [re]engineered business or application.
 - Scope of the business or application.
 - Boundaries of the business or application.

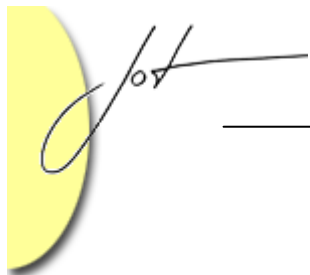


- Provide input (e.g., number of use cases and use case paths) to the endeavor's cost and schedule estimation tasks.
- Provide a basis (e.g., use cases and use case paths) for the scheduling of the development cycle phases and builds.
- Maximize the quality of the requirements (e.g., correctness, completeness, consistency, testability, and understandability).
- Maximize the productivity of the requirements teams (e.g., reuse of requirements, reuse of conventions, and existence of examples).

Requirements Engineering Tasks

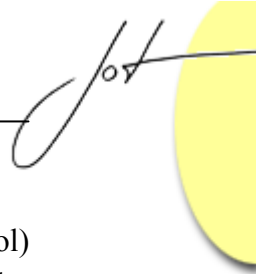
As a cohesive collection of tasks, requirements engineering typically involves the following teams and roles performing the following requirements tasks in an iterative, incremental, parallel, and time-boxed manner:

- **Business Strategy Team**, which typically performs the following tasks for all or part of a business enterprise during the business strategy or initiation phases of an endeavor:
 - **Stakeholder profiling** is the task during which the representatives of all major stakeholders of customer organization's current business enterprise of the are studied, modeled, and analyzed.
 - **Customer analysis** is the task during which the current business enterprise of the customer organization is studied, modeled, and analyzed.
 - **Competitor analysis** is the task during which competing businesses of the current business enterprise of the customer organization are identified, profiled, studied, and analyzed.
 - **Market analysis** is the task during which the current or planned marketplaces in which the business enterprise of the customer organization are identified, studied, modeled, and analyzed.
 - **User analysis** is the task during which the current and future intended user organizations of the application(s) of the customer organization's business enterprise are identified, studied, modeled, and analyzed.
 - **Business visioning** is the task during which the customer organization's vision of their [re]engineered business enterprise is produced and documented.
 - **Requirements elicitation** is the task during which raw new potential requirements for the business enterprise are identified and captured.
 - **Requirements analysis** is the task during which elicited and reused requirements for the business enterprise are studied, modeled, refined, prioritized, scheduled, and traced.
 - **Requirements specification** is the task during which requirements, requirement diagrams, and requirements models for the business enterprise are documented in requirements specifications and related documents.



- **Requirements reuse** is the task during which reusable requirements and requirements-related analyses are identified, evaluated for relevancy, and where appropriate reused (possibly with modification).
- **Requirements management** is the task during which the storage, access, approval, publication, and tracing of requirements work products are managed.
- **Technology Strategy Team**, which typically performs the following tasks during the business strategy or initiation phases of an endeavor:
 - **Technology analysis** is the task during which the potential technologies for future applications are identified, analyzed, and documented.
 - **Requirements reuse** (see above).
 - **Requirements management** (see above).
- **Requirements Team**, which typically performs the following tasks for a single application, component, or center during the initiation and construction phases of an endeavor:
 - **Requirements visioning** is the task during which the customer organization's vision of a single application, component, or center is produced and documented.
 - **Requirements elicitation** is the task during which raw new potential requirements for the application, component, or center are identified and captured.
 - **Requirements prototyping** is the task during which one or more prototypes are produced in order to identify and iterate requirements.
 - **Requirements analysis** is the task during which elicited and reused requirements for the application, component, or center are studied, modeled, refined, prioritized, scheduled, and traced.
 - **Requirements specification** is the task during which requirements, requirement diagrams, and requirements models for the application, component, or center are documented in requirements specifications and related documents.
 - **Requirements reuse** (see above).
 - **Requirements management** (see above).
- **Architecture Team**, which typically performs the following task during the initiation and construction phases of an endeavor:
 - **Requirements analysis** is the task during which they help prioritize the requirements from the implementability and testability standpoints.
- **Customer representatives**, which typically perform the following task during the initiation and construction phases of an endeavor:
 - **Requirements analysis** is the task during which they help prioritize the requirements from the business and marketing standpoints.

Although scope management is primarily a management (rather than a requirements engineering) task, it does heavily involve requirements and could have been added to the



above list. Similarly, there are numerous quality engineering (i.e., assurance and control) tasks involving requirements engineering and its associated requirements work products.

Requirements Engineering Techniques

A **technique** is a low-level work unit that is the reified implementation of (i.e., way of performing) one or more tasks. The following techniques have been found to be useful when performing requirements engineering:

- **Abstraction** can be used to produce requirements and other work products containing the correct level of detail.
- **Brainstroming** can be used to rapidly identify stakeholders and elicit requirements.
- **Cohesion** can be used to produce and evaluate requirements and other requirements work products.
- **Cost/benefit analysis** can be used to determine whether requirements and their associated applications are going to be cost-effective.
- **Cost estimation** can be used to estimate the cost of implementing or making changes to requirements.
- **Coupling** can be used to minimize the interrelationships between requirements and other requirements work products.
- **Cross functional teams** can be used to perform requirements engineering in a manner that guarantees inputs from all stakeholders.
- **Documentation standards** (e.g., content and format standards) can be used create requirements specifications.
- **Documentation studies** can be used to identify requirements from existing documents.
- **Documentation templates** can be used to generate skeleton requirements specifications.
- **Gap analysis** can be used to analyze the difference between the capabilities of the current application and the needs of future versions of the application.
- **Incremental development** can be used to incrementally elicit, analyze, and specify the requirements.
- **Inspection checklists** can be used to evaluate the quality of the requirements work products.
- **Interviews** can be used to elicit requirements from stakeholders.
- **Iteration** can be used to improve the quality of the requirements work products.
- **Joint application development (JAD)** can be used to elicit and analyze the requirements with the stakeholders, especially the customer representatives.
- **Language standards** can be used to ensure the understandability and value of requirements modeling and specification languages.
- **Observation** can be used to determine user needs and requirements.

- **Parallel development** can be used to concurrently develop the different requirements work products and to develop them concurrently with the architecture and associated system tests.
- **Prototyping** can be used to prototype the user interface and thereby elicit and analyze user requirements.
- **Questionnaires** can be used to elicit stakeholder needs and requirements.
- **Reference requirements** can be used as standard, common, reusable requirements across multiple endeavors.
- **Requirements patterns** can be used to find standardized solutions to commonly occurring problems within requirements engineering.
- **Requirements standards** can be used to set the content, format, and minimal quality of requirements work products, especially requirements specifications.
- **Storyboarding** can be used to elicit and understand user requirements.
- **Timeboxing** can be used to ensure that each requirements engineering task is completed in a reasonable time frame and thereby to avoid “analysis paralysis.”

2 REQUIREMENTS-RELATED PRODUCERS

There are numerous work products produced during requirements engineering, and these work products do not produce themselves. Thus requirements engineering involves producers, which may be categorized as roles, teams, and organizations, as well as the tools that they use. The following simplified figure illustrates the different kinds of producers and how they relate to work products and work units:

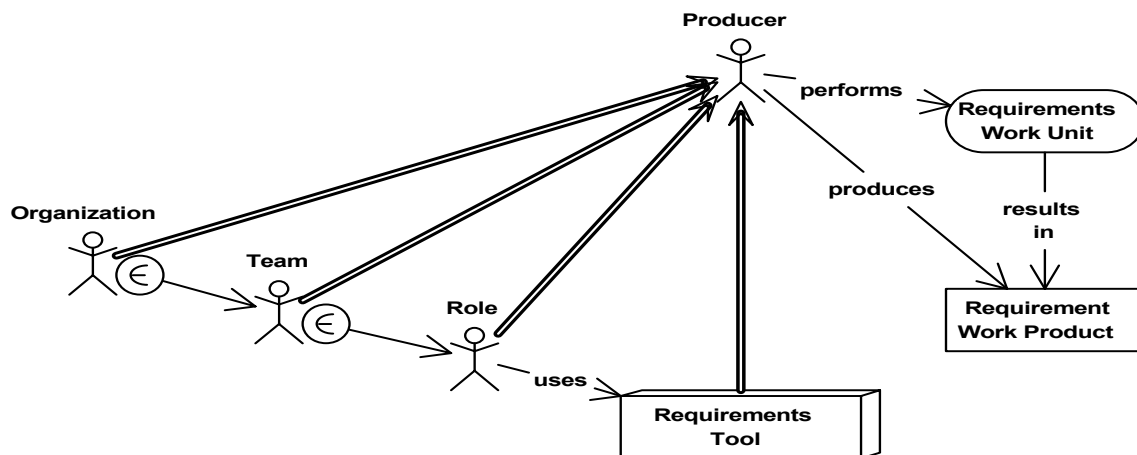


Fig. 2: Requirements Producers

Roles

A **role** is a low-level producer that represents a part played by (i.e., a cohesive collection of responsibilities of) a person on an endeavor. The following roles perform tasks of (or highly related to) the requirements engineering activity:

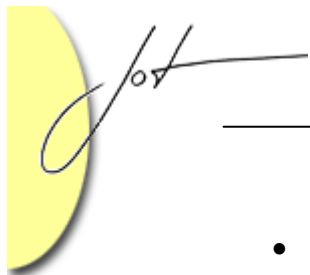


- **Business architect**, who rearchitects the business by creating new business models, processes, organizational structures, and recommends new applications.
- **Business strategist**, who leads the business strategy team, analyzes the customer organization, analyzes the market, and produces the business vision.
- **Customer representative**, who helps produce the business vision, provides requirements, and prioritizes the requirements from a business perspective.
- **Digital brand strategist**, who produces any requirements relating to digital branding.
- **Domain expert** (in the business domain, market, and application domain), who provides requirements, verifies the analysis, and provides recommendations regarding the business and application vision statements.
- **Process engineer**, who helps produce the requirements engineering process and evaluates the actual performance of the process.
- **Project manager**, who evaluates, approves, and manages the scope of the requirements.
- **Requirements engineer**, who leads the requirements team and performs most of the requirements tasks.
- **Security analyst**, who helps elicitate, analyze, and specify security requirements.
- **Software architect**, who helps prioritize requirements from an implementation perspective.
- **System architect**, who helps prioritize requirements from an implementation perspective.
- **Technical leader**, who evaluates and prioritizes the requirements from an implementation standpoint.
- **Technical writer**, who produces and maintains the requirements documents.
- **Technology strategist**, who analyzes the relevant technologies and their trends and helps establish any technology constraints.
- **Test engineer**, who ensures that all requirements are validatable (e.g., testable) and who acts as a liaison to the various teams performing testing.
- **User analyst**, who analyzes users and user organizations.
- **User representative**, who helps analyze the user organizations and provides requirements from a user perspective.

Teams

A **team** is a relatively small-scale producer with the following characteristics:

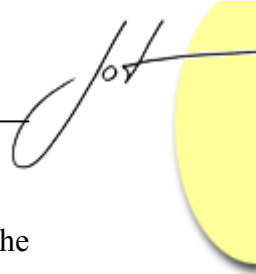
- **Structure**. A team consists of a cohesive collection of one or more related roles and/or subordinate teams that collaborate to perform a cohesive set of tasks (i.e., typically an activity, although the tasks may comprise a major subset of an activity or a set of related activities).
- **Parent producer**. A team is a major component of an organization.
- **Endeavor**. A team is a major component of a program or project.



- **Management.** A team is managed as a unit.

The following teams typically produce the requirements work products and perform the requirements-related work units:

- **Business strategy team**, which typically produces and documents the new business strategy (including the business vision and list of applications to produce or upgrade) for the customer organization's business enterprise by analyzing the customer organization, the marketplace in which the customer organization competes, and the associated user organizations. It should typically consist of people playing one or more of the following roles:
 - Business Architect
 - Business Strategist (team leader)
 - Customer Representative
 - Digital Brand Strategist (if relevant)
 - Domain Expert (in the business domain and market)
 - Technical Writer
 - Requirements Engineer
 - User Analyst
 - User Representative
- **Technology strategy team**, which typically produces and documents the new technology strategy (including selecting major common technologies) for the customer organization's business enterprise by analyzing technologies, technology trends, and the technologies currently used by the business. It should typically consist of people playing one or more of the following roles:
 - Technology Strategist (team leader)
 - Security Analyst
 - Technical Writer
- **Requirements team**, which typically performs requirements engineering tasks for an application, application domain, framework, component, or (contact or data) center during the initiation and construction phases of an endeavor. It should typically consist of people playing one or more of the following roles:
 - Customer Representative
 - Domain Expert
 - Requirements Engineer (team leader)
 - Security Analyst
 - Software Architect
 - System Architect
 - Technical Writer
 - Test Engineer
 - User Representative
- **Strategy inspection team**, which is the team that inspects the deliverable business requirements and architecture work products that are produced by the



- strategy teams. It should typically consist of people playing on or more of the following roles:
- Business Architect
 - Independent Business Strategist
 - Customer Representative
 - Independent Digital Brand Strategist (if relevant)
 - Independent Domain Expert
 - Process Engineer
 - Project Manager
 - Quality Engineer
 - Requirements Engineer
 - System Architect
 - Technical Leader (team leader)
 - Technical Writer
 - Test Engineer
 - **Requirements inspection team**, which is the team that inspects the deliverable requirements work product produced by the requirements team . It should typically consist of people playing on or more of the following roles:
 - Business Strategist
 - Customer Representative
 - Database Architect
 - Domain Expert
 - Hardware Architect
 - Metrics Analyst
 - Process Engineer
 - Project Manager
 - Quality Engineer
 - Independent Requirements Engineer
 - Security Analyst
 - Security Architect
 - Software Architect
 - System Architect
 - Technical Leader (team leader)
 - Technical Writer
 - Test Engineer
 - User Analyst
 - **Architecture team**, which evaluates the requirements and helps prioritize them from an implementation standpoint.
 - **Management team**, which approves the requirements, manages the scope of the endeavor in terms of the requirements, staffs the requirements teams, and schedules the endeavor (partially in terms of delivered requirements).

- **Quality team**, which performs quality control tasks on the requirements work products and quality assurance tasks on the requirements engineering process.

Tools

Requirements engineering on non-trivial endeavors involves the analysis and management of large numbers of requirements and other requirements work products including diagrams, models, and documents. The following tools are used to support the requirements engineering tasks:

- **Documentation tools**, which are used to produce the requirements specifications and other documents, either manually or automatically via the information stored in either the upperCASE tools or requirements management tools.
- **Configuration management tools**, which are used to control access to baselined requirements and other requirements work products.
- **Requirements management tools**, which are used to enable requirements teams (and other stakeholders) to:
 - Store requirements.
 - Access requirements (e.g., directly or via filter or query).
 - Prioritize requirements.
 - Store characteristics (attributes or metadata) of requirements.
 - Store relationships between requirements.
 - Trace requirements.
 - Identify requirements baselines, and control versions of and changes to the requirements (i.e., configuration management).
 - Automate requirements specification production and publication.
 - Obtain notifications when relevant requirements are changed (e.g., via publish/subscribe).
 - Free up scarce human resources for higher level and more valuable tasks by automating the preceding manual, record keeping parts of the requirements management task.
- **UpperCASE tools**, which are used to perform requirements analysis tasks (e.g., making requirements diagrams and models).

Organizations

An organization is a relatively large-scale indirect producer with the following characteristics:

- **Endeavor.** An organization is a major component of an enterprise.
- **Structure.** An organization consists of a cohesive collection of one or more related teams (and suborganizations).
- **Management.** An organization is managed as a unit.

Essentially all organizations are involved with requirements engineering including:

- **Customer Organization**, which supplies and approves the requirements.



- **Development Organization**, which engineers and implements the requirements.
- **Maintenance Organization**, which supplies some of the requirements.
- **Marketing Organization**, which supplies some of the requirements.
- **Operations Organization**, which supplies some of the requirements.
- **Partner Organization**, which supplies some of the requirements.
- **Reuse Organization**, which supplies reusable requirements and other requirements work products.
- **Subcontractor Organization**, which implements the requirements.
- **Support Organization**, which supplies some of the requirements.
- **User Organization**, which supplies some of the requirements.
- **Vendor Organization**, which supplies products which must implement requirements.

3 REQUIREMENTS LANGUAGES

A language is set of terms combined with associated syntax and semantic rules that are used to produce one or more work products. Whereas programmers are primarily concerned with various kinds of implementation languages, requirements engineers and their technical writers use the following kinds of languages to implement requirements work products:

- **Natural languages**, such as English, are most often used to specify textual requirements, although natural languages suffer numerous problems such as ambiguity that can make them difficult to understand and use.
- **Modeling languages**, such as the Unified Modeling Language (UML) and the Object Modeling Language (OML), can be used to produce requirements diagrams and associated graphical models.
- **Specification languages**, such as Z (pronounced Zed) and Object-Z, are sometimes used to specify requirements more formally than they can be specified using natural languages.

4 CONCLUSION

As the preceding two articles shows, requirements engineering is not trivial and can involve a great many reusable process components that must be selected, integrated, and tailored to produce a complete project-specific requirements process. Future articles in this column will cover individual requirements-related process components of the OPEN Process Framework such as quality requirements, requirements engineering tasks, requirements tools, and requirements engineering teams. For more detailed information about the OPEN Process Framework, go to the official OPEN website at www.open.edu or my website at www.donald-firesmith.com.

The next article in this column will discuss the various kinds of security requirements, how to specify them, and how to avoid unnecessarily specifying architectural security mechanisms as either requirements or constraints.

REFERENCES

[Firesmith2001] Donald Firesmith and Brian Henderson-Sellers: *The OPEN Process Framework*, Addison-Wesley-Longman, 2001.

About the author



Donald Firesmith runs Firesmith Consulting, which provides consulting and training in the development of software-intensive systems. He has worked exclusively with object technology since 1984 and has written 5 books on the subject. Most recently, he has developed a 1000+ page informational website on the OPEN Process Framework. His most recent book is *The OPEN Process Framework* (Addison-Wesley-Longman, 2001), and he is currently writing a book on requirements engineering. He can be reached at donald_firesmith@hotmail.com.