Glossary

Actor The person or automated system that interacts with a product use case. Actors are also known as *users* and end users.

Adjacent system A system (person, organization, computer system) that provides information to, or receives information from, the work that you are studying.

Agile development A way of developing software using *iterative development*. A variety of agile techniques exist, including Scrum, eXtreme Programming, and Crystal. We use the term *iterative* to mean any agile or iterative technique.

Agile Manifesto A set of principles that focuses on the delivery of working software to the customer, collaborative working practices, and the ability to respond rapidly to change.

Atomic requirement A requirement that is testable and does not need further decomposition. See also **Functional requirement**, **Non-functional requirement**, **Constraint**, **Snow card**.

Attribute A data element that describes (and has been attributed) to a class. Also used to mean the components of an *atomic requirement*.

Blastoff A technique for building the foundation for the requirements activity by establishing the scope–stakeholders–goals trinity, and verifying the viability of the project.

Brown Cow Model A technique for discussing and modeling the work by using different system viewpoints. The viewpoints are permutations of now, future, how, and what.

Business analyst The role responsible for discovering and communicating requirements among the project's stakeholders. Also referred to as requirements analyst, requirements engineer, and systems analyst.

Business data model A model of the essential classes of business data and associations between them necessary to support the functional requirements within the scope of the work.

Business event Something that happens to the business (usually called "the work"), that makes it respond. Examples: "Customer pays an invoice," "Truck reports all roads have been treated," "Time to read electricity meters," "Surfer wants to search website."

Business process model A model of the processes within a business; often used to understand the current processes so that they can be improved. A variety of notations can be used for such models.

Business use case (BUC) The work's response to a business event. It includes the processes and the stored data needed to satisfy the request implicit in the business event. See also **Product use case**.

Class A physical or abstract entity within a context of study that has one or more attributes of stored data.

Client The person who pays for the development of the product, or who has organizational responsibility for the project. Also known as the *sponsor*.

Constraint A requirement, either organizational or technological, that restricts the way you produce the product. It may be a management edict on the way the product must be designed—"It must work on a 4G mobile phone"—or a budget that limits the extent of the product or a characteristic of the current technology that constrains the possible solutions.

Context The subject matter, people, and organizations that affect the requirements for the product. The context of study, or the work context, identifies the business being studied and the adjacent systems that interact with this work. The product context identifies the scope of the product and its interactions with users and other actors.

Context diagram A graphic model of the *work scope* showing how the work to be investigated connects to the outside world.

Commercial off-the-shelf (COTS) products Products (usually software) developed by external organizations to carry out a specified range of functionality.

Customer A person who buys the product.

Data dictionary Specification (to the elemental level) of the terms used in the requirements specification.

Data element A piece of data that has a defined range or set of values within a context of study. Also called an *attribute*.

Data flow Data that moves from one process to another; usually represented by a named arrow.

Data model A model showing *classes* of data and the associations between them. This is also known as a class model. See also **Business data model**.

Design The act of crafting a technological solution to fit the requirements, within the constraints.

Developer Someone who is part of the technological development or building of the product. Examples: programmer (often called a developer) designer, architect.

Domain analysis The activity of investigating, capturing, and specifying generic knowledge about a subject-matter area.

Essential viewpoint A viewpoint that focuses on the rules, policy, and data independent from any implementation of those rules, policy, and data. Also referred to as *thinking above the line*.

External event An event triggered by a happening in an adjacent system outside the work context. Example: “Customer wants to open an account.” See also **Time-triggered event**.

Fit criterion A quantification or measurement of the requirement such that you are able to determine whether the delivered product satisfies (fits) the requirement.

Function point A measure of the functionality of the work or a piece of software. Function points were first proposed by Allan Albrecht; today the method for counting function points is specified by the International Function Point User Group.

Functional requirement Something that the product must do. Functional requirements are part of the fundamental processes of the product.

Innovation Fresh thinking, or thinking differently, about the problem to find a new and better way to do the work.

Iterative development Strategies that facilitate repetitive and continuous delivery of software solutions.

Naming conventions The terminology (including abbreviations and acronyms) that the stakeholders use when talking about the work. This terminology is usually captured in a glossary such as the one you are reading.

Non-functional requirement A property or quality that the product must have, such as an appearance, speed, security, or accuracy property.

Persona A virtual person, archetypical of your users, invented to help you to discover the requirements.

Product The thing you are writing your requirements for. In this book, “product” usually means a software product, but the requirements can be for hardware, consumer goods, a service, or any other thing you need to specify.

Product use case (PUC) The part of the business use case you decide to automate. You write the requirements for the product use case. See also **Business use case**.

Project blastoff See **Blastoff**.

Project goal The reasons for doing the project. The goal must include a quantification of the expected benefit.

Prototype A simulation of the product using software prototyping tools, low-fidelity whiteboards, or paper mockups. Also referred to as a *sketch*.

Quality Gateway Application of a set of tests (relevance, ambiguity, viability, fit, and so on) to assure the quality of individual requirements before the requirements become part of the requirements specification.

Rationale The justification for a requirement. It is used to help understand a requirement, and usually reveals the underlying reason for the requirement.

Requirement Something that the product must do, or a property that the product must have, that is needed or wanted by the stakeholders.

Requirements analyst The person who has responsibility for producing the requirements specification. The analyst does not necessarily do all of the requirements elicitation, but is responsible for coordinating the requirements effort. Depending on how roles are defined in an organization, this individual might be referred to as a *business analyst*, systems analyst, or requirements engineer.

Requirements creep The uncontrolled addition of new requirements to the product after the requirements are considered complete.

Requirements knowledge model A conceptual filing system—usually implemented as a data model—that establishes a common language for communicating requirements. (An example appears in **Appendix D**.)

Requirements pattern A cohesive collection of requirements that carry out some recognizable and potentially recurring functionality.

Requirements specification A complete collection of requirements knowledge for a specific project. The specification defines the product and may be used as a contract to build the product.

Requirements specification document A document that contains all or part of the requirements specification knowledge, depending on who and why the knowledge is being communicated.

Requirements specification template A guide for gathering and organizing requirements knowledge. (An example appears in **Appendix A**.)

Requirements tool A software tool capable of maintaining all or part of the requirements specification.

Retrospective A review designed to gather experience and provide input into improving the requirements process. Also known as "lessons learned."

Scenario A breakdown of a business use case, or a product use case, into a series of stakeholder-recognizable steps. Scenarios are used for discovering and communicating work knowledge.

Sketch A quick and dirty model or *prototype* of a piece of work or a proposed product. Sketches can be created on any medium—paper, electronic, whiteboard—and are intended to make it easier for stakeholders to understand and discuss their requirements.

Snow card A paper card showing the attributes of an atomic requirement. Most requirements teams use an electronic version of the snow card.

Solution A way of implementing the requirements. Also called the *product*.

Sponsor See **Client**.

Stakeholder Any person who has an interest in the product and who therefore has requirements for it, such as the client, a user, and someone who builds the product. Some stakeholders are remote, such as an auditor, a safety inspector, and the company lawyer.

System In the context of this book, a business or work system, and not just a computer or software system.

Systemic thinking The technique of understanding how things affect one another within a whole.

Systems analysis The craft of modeling the system's functions and data. Systems analysis can be done in several ways: data flow modeling, as defined by DeMarco; event response modeling, as per McMenamin and Palmer; use cases, as per Jacobson; or any of the many object-oriented methods, most of which use the Unified Modeling Language notation.

Technological requirement A requirement that is necessary only because of the chosen technology; it is not there to satisfy a business need.

Thinking above the line Deriving and discussing the essence of the work. The "thinking" refers to an exploration of the possibilities without being constrained (for the moment) by technological realities. The "line" is the horizontal line on the *Brown Cow Model* that separates the physical reality (how) from the essential policy (what).

Time-triggered event An event triggered by some time-related policy. Examples: "Time to produce sales report," "Time to remind driver to renew license." See also **External event**.

Trawling techniques Techniques for discovering, eliciting, determining, and innovating requirements.

Use case A chunk of functionality describing the interaction between a *user/actor* and an automated system. See also **Business use case** and **Product use case**.

User The person or system that uses the product to do work. Also known as an *actor* or end user.

User story A technique for discovering requirements by using the guide: “As a [. . .] I want [. . .] so that [. . .].” Sometimes called a story card.

Volere A set of principles, processes, templates, tools, and techniques developed to improve the discovery, communication, and management of requirements. (*Volere* is the Italian verb for “to wish” or “to want.”)

Work A business area of the organization that your product is meant to improve. The business analyst studies the work before making decisions on the optimal product to do some or all of the work.

Work scope The extent of the business area under investigation, and the real world that surrounds it. It is usually shown using a *context diagram*.