Requirement

Table of Contents

Business Analyst ***1***

Software Development Models ***2***

Iterative & Incremental **2**

RAD 3

Agile 3

SCRUM 4

A requirements development process framework **5**

Business Analyst

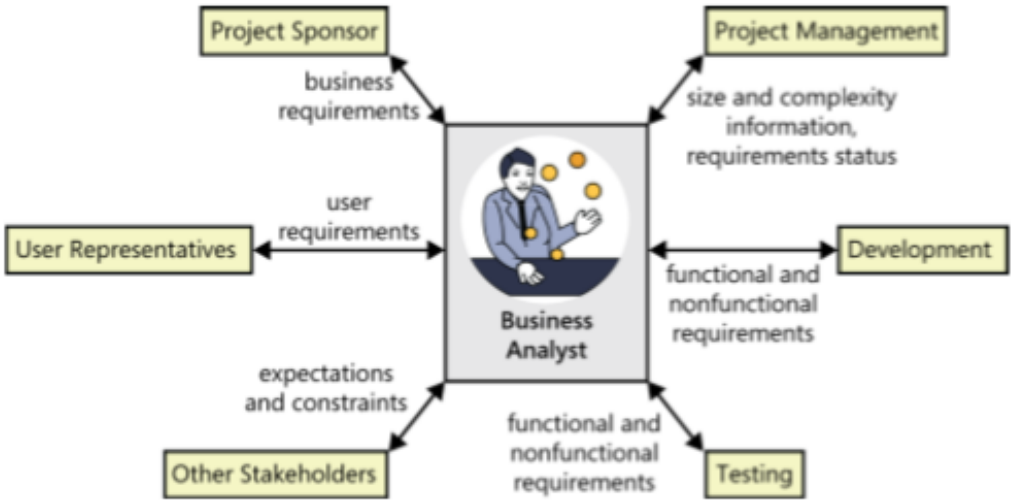


FIGURE 4-1 The business analyst bridges communication between customer and development stakeholders.

Term	Definition
BA (Business Analyst, Requirement Engineer)	The role on a project team that has primary responsibility for working with stakeholder representatives to elicit, analyze, specify, validate, and manage the project's requirements. Also called a requirements analyst, system analyst, requirements engineer, requirements manager, business systems analyst, and simply analyst.
Baseline	a snapshot in time that represents the current agreed-upon, revired, approved set of requirements.
Acceptance criteria	conditions that product must satisfy to be accepted by user, customers or related stakeholders

Rights that customers can expect when it comes to requirements issues

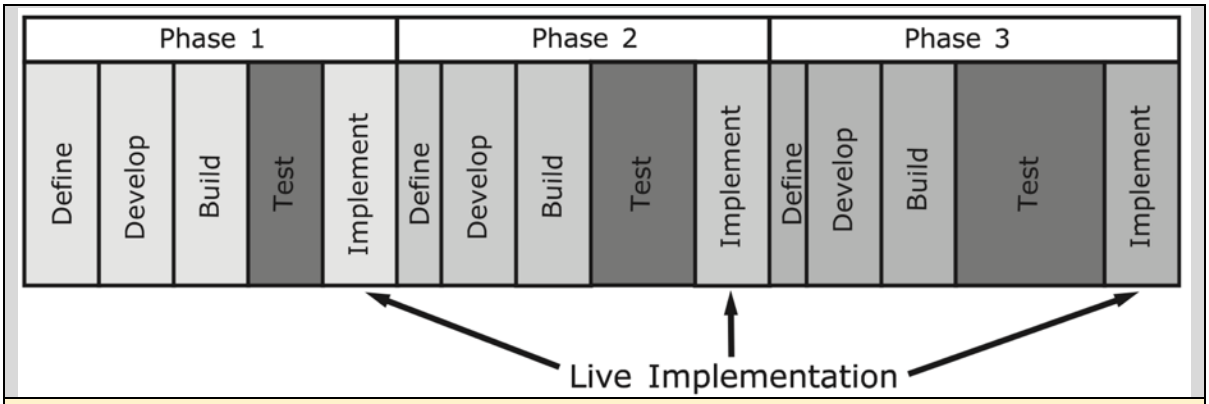
- Mutual respect
- To expect BAs to speak your language
- To expect BAs to learn about your business and your objectives
- To expect BAs to record requirements in an appropriate form
- To receive explanations of requirements practices and deliverables
- To hear alternatives / ideas from BA
- Describe characteristics that will maje the product easy to use
- To receive a system that meets your functional needs and quality expectations
- To hear about ways to adjust requirements to accelerate development through reuse
- You have the right to make changes in the requirements as the business evolves, as the team gathers more input from stakeholders, or as you think more carefully about what you need.

BA's tasks

- Define business requirements
- Plan the requirements approach
- Identify project stakeholders and user classes
- Elicit requirements
- Analyze requirements
- Document requirements
- Communicate requirements
- Lead requirements validation
- Facilitate requirements prioritization
- Manage requirements

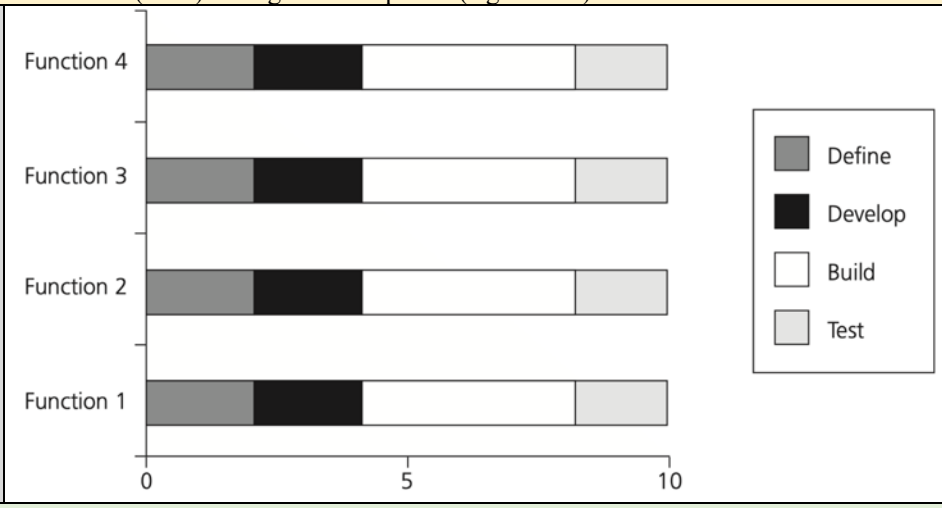
Software Development Models

Iterative & Incremental
Iterative development model A development lifecycle where a project is broken into a usually large number of iterations. An iteration is a complete development loop resulting in a release (internal or external) of an executable product, a subset of the final product under development, which grows from iteration to iteration to become the final product.
Incremental development model A development lifecycle where a project is broken into a series of increments, each of which delivers a portion of the functionality in the overall project requirements. The requirements are prioritized and delivered in priority order in the appropriate increment. In some (but not all) versions of this lifecycle model, each subproject follows a 'mini V-model' with its own design, coding and testing phases



Examples of iterative or incremental development models are prototyping, Rapid Application Development (RAD), Rational Unified Process (RUP) and agile development (e.g. Scrum).

RAD



Components/functions are developed in parallel as if they were mini projects, the developments are time-boxed, delivered, and then assembled into a working prototype. This can very quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements. Rapid change and development of the product is possible using this methodology. However the product specification will need to be developed for the product at some point, and the project will need to be placed under more formal controls prior to going into production. This methodology allows early validation of technology risks and a rapid response to changing customer requirements

Agile

A group of software development methodologies based on iterative incremental development, where requirements and solutions evolve through collaboration between self-organizing cross- functional teams

Agile manifesto: A statement on the values that underpin agile software development.

1. individuals and interactions over processes and tools
2. working software over comprehensive documentation
3. customer collaboration over contract negotiation
4. responding to change over following a plan

While there are a number of agile methodologies in practice, the industry seems to have settled on the use of **Scrum** as an agile management approach, and **Extreme Programming (XP)** as the main source of agile development ideas

Some characteristics of project teams using Scrum and XP

1. The generation of **business stories** (a form of lightweight use cases) to define the functionality, rather than highly detailed requirements specifications.
2. The incorporation of **business representatives** into the development process, as part of each iteration (called a '**sprint'** and typical lasting 2 to 4 weeks), providing continual feedback and to define and carry out functional acceptance testing.

3. The **recognition** that we can't know the future, so changes to requirements are welcomed throughout the development process, as this approach can produce a product that better meets the stakeholders' needs as their knowledge grows over time.
4. The concept of **shared code ownership** among the developers, and the close inclusion of testers in the sprint teams.
5. The **writing of tests** as the first step in the development of a component, and the automation of those tests before any code is written. The component is complete when it then passes the automated tests. This is known as **Test-Driven Development**.
6. **Simplicity**: building only what is necessary, not everything you can think of.
7. The **continuous integration and testing** of the code throughout the sprint, at least once a day.

SCRUM	a framework for managing and controlling iterative projects where the product owner works with cross-functional teams to create a list of tasks to be done. **Scrum is based on the theory of empirical process control, which relies on transparency, inspection, & adaptation. Scrum Is Iterative & Incremental** - **Transparency:** ***Scrum Reviews Provide Transparency.*** - **Inspection:** *Scrum Reviews & Retrospectives Offer Inspection Opportunities.* - **Adaptation:** *Scrum Teams Can Adapt the Product at the End of Every Sprint.*

5 Values of SCRUM

- **Commitment:** The Scrum value of commitment is essential for building an agile culture. Scrum teams work together as a unit (Scrum and agile teams trust each other to follow through on what they say they are going to do.)
- **Courage:** The Scrum value of courage is critical to an agile team's success. Scrum teams must feel safe enough to say no, to ask for help, and to try new things. Agile teams must be brave enough to question the status quo when it hampers their ability to succeed.
- **Focus: T**he Scrum value of focus is one of the best skills Scrum teams can develop. Focus means that whatever Scrum teams start they finish--so agile teams are relentless about limiting the amount of work in process (limit WIP).
- **Openness:** Scrum teams consistently seek out new ideas and opportunities to learn. Agile teams are also honest when they need help.
- **Respect:** Scrum team members demonstrate respect to one another, to the product owner, to stakeholders, and to the ScrumMaster. Agile teams know that their strength lies in how well they collaborate, and that everyone has a distinct contribution to make toward completing the work of the sprint. They respect each other's ideas, give each other permission to have a bad day once in a while, and recognize each other's accomplishments.

Terms of SCRUM

- **Sprint**: An uninterrupted development period, typically one to four weeks in duration, during which a development team implements a defined set of functionality selected from the product backlog or baselined requirements for the product.
- **SCRUM team**: collection of individuals (typically between five and nine members) working together to deliver the required product increments.
- **Product Owner**: A role, typically on an agile project team, that represents the customer and that is responsible for setting the product vision, providing project boundaries and constraints, prioritizing the contents of the product backlog, and making product decisions.
- **Development Team**: are the people that do the work. The development team can be comprised of all kinds of people including designers, writers, programmers, etc.
- **Scrum Master**: expert in SCRUM to guide the team through the scrum process with their experience and expertise.
- **Product Backlog**: On an agile project, the prioritized list of work remaining for the project. A backlog can contain user stories, business processes, change requests, infrastructure development, and defect stories. Work items from the backlog are allocated to upcoming iterations based on their priority.
- **Sprint Backlog:** is the set of items that a cross-functional product team selects from its product backlog to work on during the upcoming sprint.
- Potentially Releasable Product Increment
- Sprint Burndown Chart

- Sprint Planning
- Daily Scrum:
- Sprint Review
- Sprint Retrospective
- **User story**: s a tool used in Agile software development to capture a description of a software feature from an end-user perspective.
- Theme
- Epic (in Jira)

A requirements development process framework

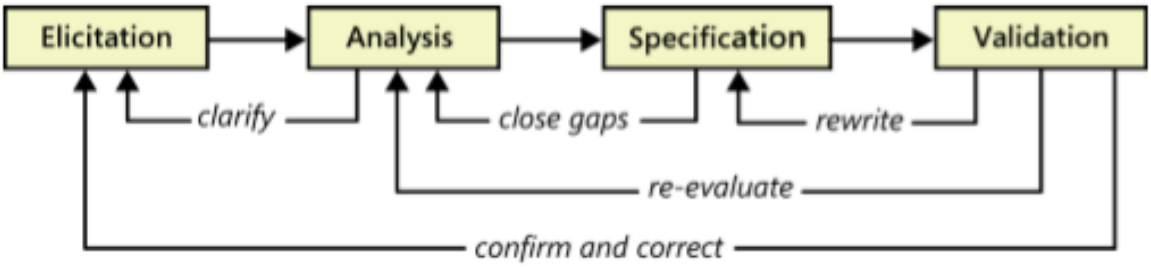


FIGURE 3-1 Requirements development is an iterative process.