1. The Essential Software Requirement

Who should learn this course? 1

- The project's business objectives, vision, and scope were never clearly defined.
- Customers were too busy to spend time working with analysts or developers on the requirements.
- Your team could not interact directly with representative users to understand their needs.
- Customers claimed that all requirements were critical, so they didn't prioritize them.
- Developers encountered ambiguities and missing information when coding, so they had to guess.
- Communications between developers and stakeholders focused on user interface displays or features, not on what users needed to accomplish with the software.
- Your customers never approved the requirements.

Who should learn this course? 2

- Your customers approved the requirements for a release or iteration and then changed them continually.
- The project scope increased as requirements changes were accepted, but the schedule slipped because no additional resources were provided and no functionality was removed.

- Requested requirements changes got lost; no one knew the status of a particular change request.
- Customers requested certain functionality and developers built it, but no one ever uses it.
- At the end of the project, the specification was satisfied but the customer or the business objectives were not.

Contents

- Software requirements defined
- Requirements development and management
- Every project has requirements
- When bad requirements happen to good people
- Benefits from a high-quality requirements process

1.1 Software requirements defined

- Interpretations of “requirement”
  - Requirements are a specification of what should be implemented.
  - They are descriptions of how the system should behave, or of a system property or attribute.
  - They may be a constraint on the development process of the system.

Levels and types of requirements

- Business requirement
- Business rule
- Constraint
- External interface requirement
- Feature

- Functional requirement
- Nonfunctional requirement
- Quality attribute
- System requirement
- User requirement

Table 1-1, page 7

Levels and types of requirements

- Business requirement A high-level business objective of the organization that builds a product or of a customer who procures it.
- Business rule A policy, guideline, standard, or regulation that defines or constrains some aspect of the business. Not a software requirement in itself, but the origin of several types of software requirements.
- Constraint A restriction that is imposed on the choices available to the developer for the design and construction of a product.
- External interface requirement A description of a connection between a software system and a user, another software system, or a hardware device.
- Feature One or more logically related system capabilities that provide value to a user and are described by a set of functional requirements.

Levels and types of requirements

- Functional requirement A description of a behavior that a system will exhibit under specific conditions.
- Nonfunctional requirement A description of a property or characteristic that a system must exhibit or a constraint that it must respect.
- Quality attribute A kind of nonfunctional requirement that describes a service or performance characteristic of a product.
- System requirement A top-level requirement for a product that contains multiple subsystems, which could be all software or software and hardware.
- User requirement A goal or task that specific classes of users must be able to perform with a system, or a desired product attribute.

Relationships among requirements information

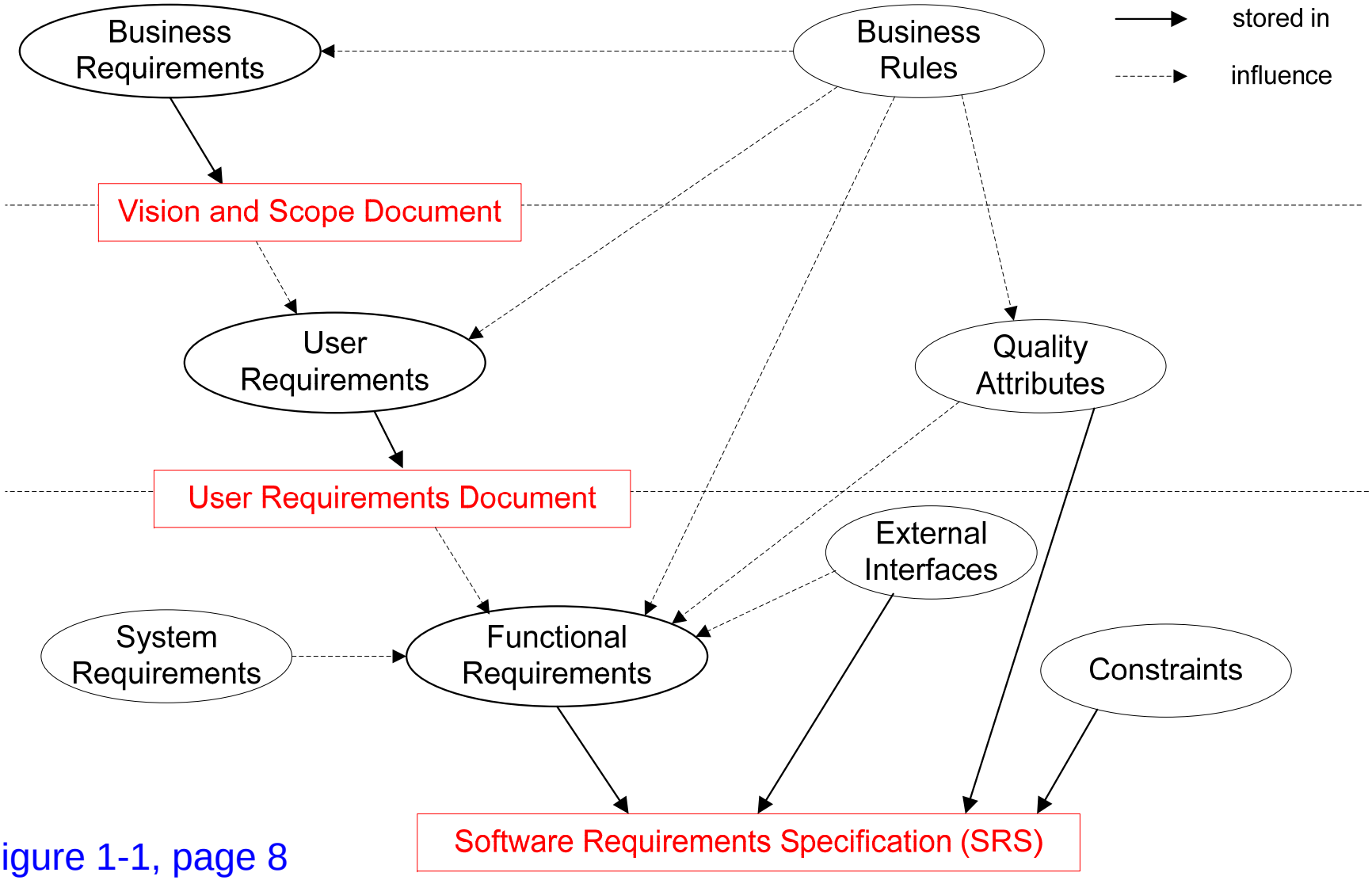


Figure 1-1, page 8

Documents in SRE

- It is important to recognize the value of recording vital requirements information in a shareable form, rather that treating it as oral tradition around the project campfire.
  - Vision and Scope Document
  - User Requirements Document
  - Software Requirements Specification (SRS)

Contents

- Software requirements defined
- Requirements development and management
- Every project has requirements
- When bad requirements happen to good people
- Benefits from a high-quality requirements process

Requirements Engineering Activities

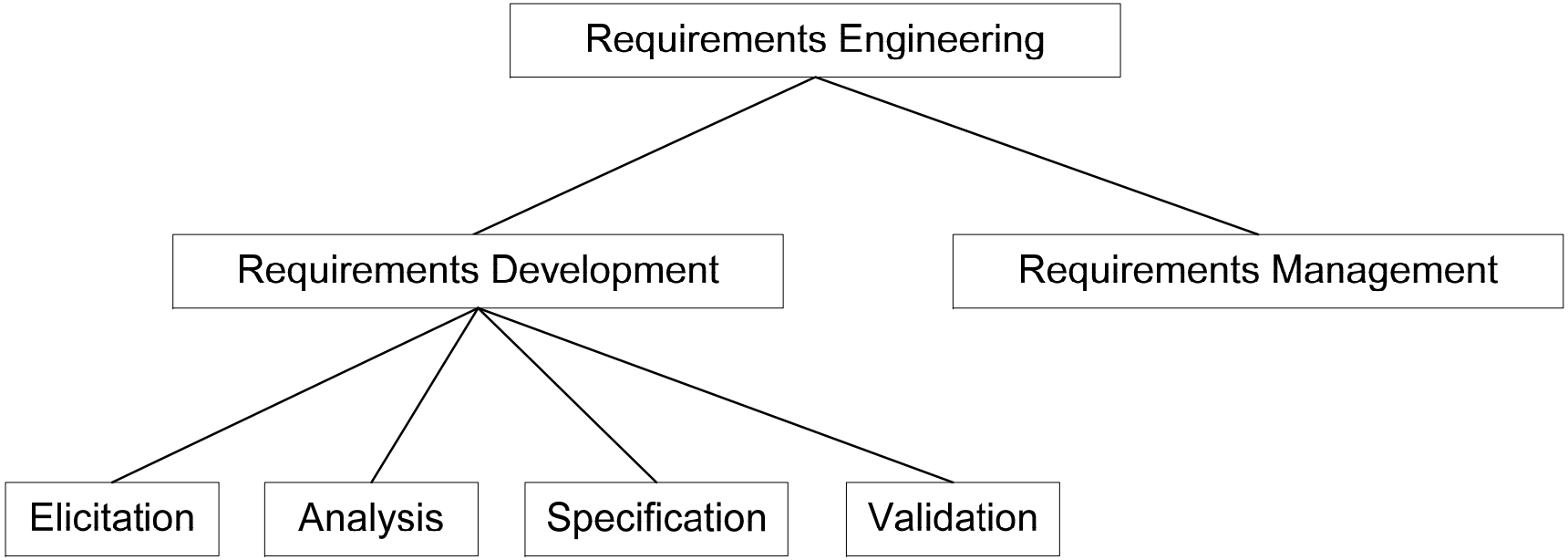


Figure 1-4, page 15

Requirements Development 1

- Elicitation
  - Indentifying the product's expected user classes and other stakeholders
  - Understanding user tasks and goals and the business objectives
  - Learning about the environment in which the new product will be used
  - Working with individuals who represent each user class to understand their functionality needs and their quality expectations

Requirements Development 2

- Analysis
  - Analyzing the information received from users to distinguish their task goals from functional requirements, quality expectations, business rules, suggested solutions, and other information
  - Decomposing high-level requirements into an appropriate level of detail
  - Deriving functional requirements from other requirements information
  - Understanding the relative importance of quality attributes
  - Allocating requirements to software components defined in the system architecture
  - Negotiating implementation priorities
  - Identifying gaps in requirements or unnecessary requirements as they relate to the defined scope

Requirements Development 3

- Specification
  - Translating the collected user needs into written requirements and diagrams suitable for comprehension, view, and use by their intended audiences.

- Validation
  - Reviewing the documented requirements
  - Developing acceptance tests and criteria to confirm

An iterative process

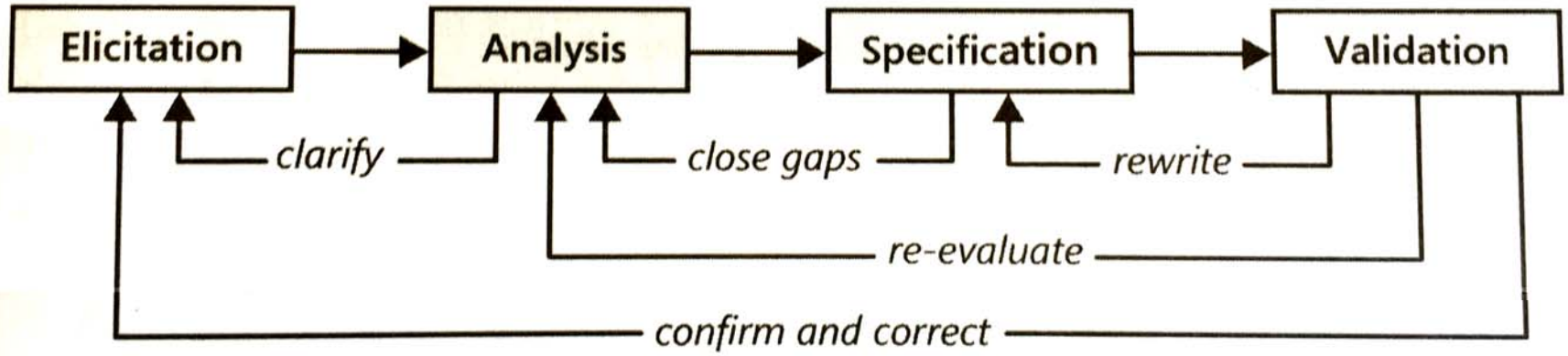
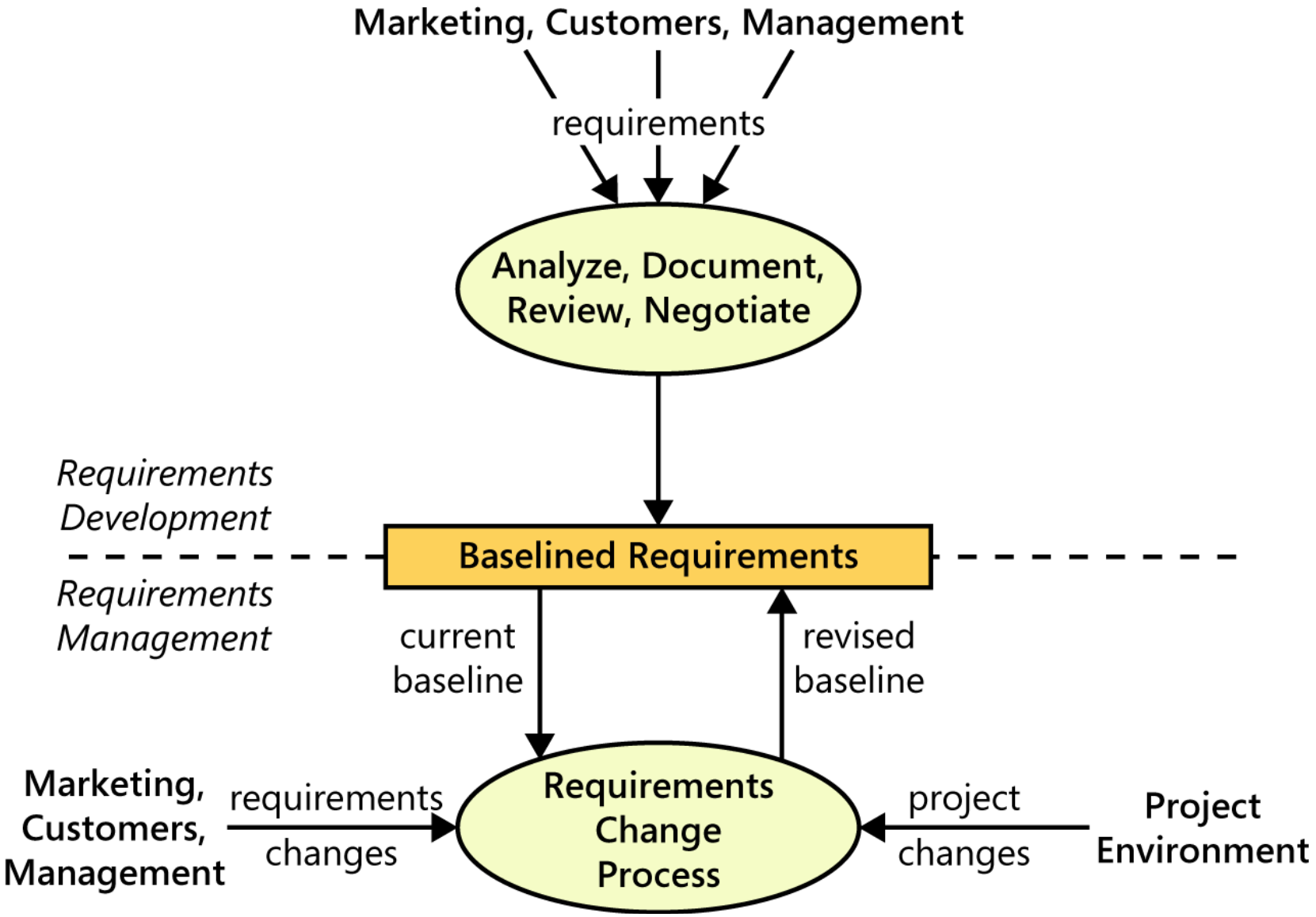


FIGURE 3-1 Requirements development is an iterative process.

Requirements Management

- Define the requirements baseline for each iteration
- Evaluate the impact of proposed changes, and incorporate approved changes into the project in a controlled way
- Keep project plans current
- Negotiate new commitments based on the estimated impact of requirements change
- Define the relationships and dependencies between requirements
- Trace individual requirements to their designs, source code, and tests
- Track requirements status and change activity throughout the project


Marketing, Customers, Management
requirements
Analyze, Document, Review, Negotiate
Requirements Development
Baselined Requirements
Requirements Management
current baseline
revised baseline
Marketing, Customers, Management
requirements changes
Requirements Change Process
project changes
Project Environment

Contents

- Software requirements defined
- Requirements development and management
- Every project has requirements
- When bad requirements happen to good people
- Benefits from a high-quality requirements process

Consequences and Causes

- The major consequence of requirements problems is rework – doing again something that you thought was already done – late in development or after release.
  - Cost, delay of delivery and money collection, reputations.

- Causes
  - Insufficient user involvement
  - Inaccurate planning
  - Creeping user requirements
  - Ambiguous requirements
  - Gold plating
  - Overlooked stakeholders

Contents

- Software requirements defined
- Requirements development and management
- Every project has requirements
- When bad requirements happen to good people
- Benefits from a high-quality requirements process

Benefits

- Reduced development rework
- Faster development and delivery
- Fewer unnecessary and unused features
- Lower enhancement costs
- Fewer miscommunications
- Reduced scope creep
- Reduced project chaos
- Higher customer and team member satisfaction
- Products that do what they’re supposed to do

Even if you can’t quantify all of these benefits, they are real.