AMDARIS

INTRODUCTION

September 2019 | Olesea Oaserele

GENERAL INFORMATION

The present course is provided by **Amdaris** company

Presenter - **Olesea Oaserele**,
Delivery Lead and Business Analyst.
olesea.oaserele@Amdaris.com

WHY THIS COURSE?

- My previous experience in Quality Assurance and Quality Analysis
- Actual hands on experience in Business and System Analysis
- Requirements Engineering are the base for the Quality Management
- Teaching is fun ☺

COURSE OBJECTIVES

- Understanding what is influencing the software quality
- Know the difference and purpose of functional and non-functional requirements.
- Understand the Quality Management throughout various SDLCs
- Be proficient in the whole requirements management process
- Know the methods of the stakeholder management
- Evaluate various ways to negotiate with the client and other stakeholders
- Be proficient in requirements specification types
- Be proficient in techniques for requirements validation
- Be able to preserve quality through change management process

COURSE STRUCTURE

- Stakeholders management
- Elicitation and Analysis
- Types of requirements
- Representing requirements
- Requirements modelling
- Requirements Validation
- Software development Lifecycle
- Quality Management
- Estimation and Prioritisation
- Change Control and preserving quality

ASSIGNEMENTS

- Pitch your idea
- Analyse your stakeholders
- Create high-level requirements
- Elicit and describe well defined requirements
- Compare techniques
- Model your requirements
- Validate your requirements
- Get Agile – create Users Stories
- Estimate the requirements and get an MVP
- Preserve the quality

What we do…

- Discuss about quality of the software from analysis and planning perspective
- Talk a lot about requirements in order to make sure quality is built in from the beginning
- Discuss about requirements engineering in different SDLCs

…and what we don’t

- Talk a lot about quality assurance in development process
- Study testing process or techniques
- Study project management

WHAT IS SOFTWARE QUALITY

And why the fuss?

DEFINITION

ISTQB – International Software Testing Qualification Board

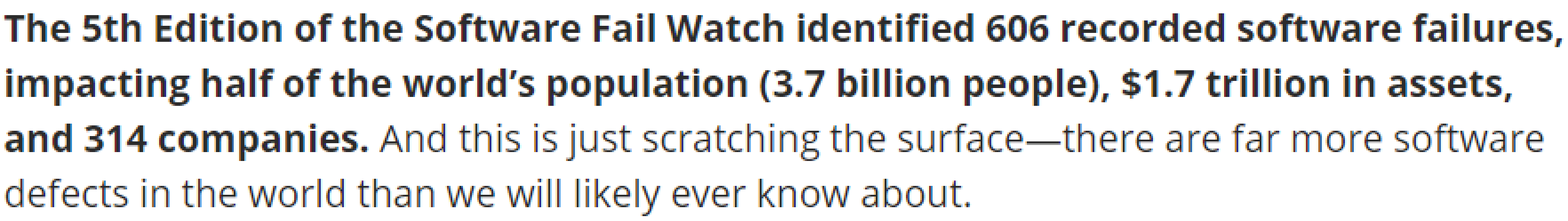
- Quality = The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations.
- Software quality: The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs.

Wikipedia

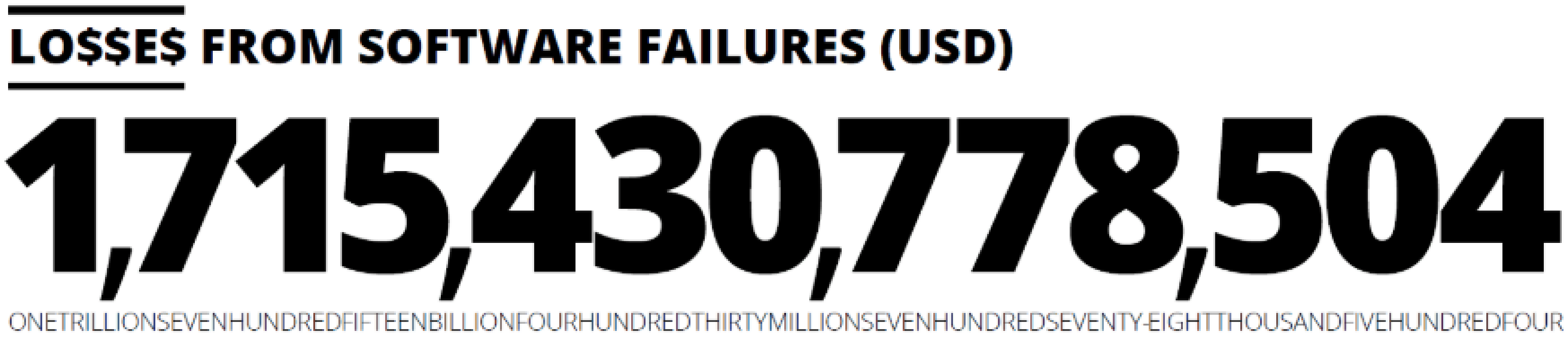
- **Software functional quality** reflects how well it complies with or conforms to a given design, based on **functional requirements** or specifications. That attribute can also be described as the fitness for purpose of a piece of software or how it compares to competitors in the marketplace as a worthwhile product. It is the degree to which the correct software was produced.
- **Software structural quality** refers to how it meets **non-functional requirements** that support the delivery of the functional requirements, such as robustness or maintainability. It has a lot more to do with the degree to which the software works as needed.

IS QUALITY SO IMPORTANT?

Yes, of course...



The 5th Edition of the Software Fail Watch identified 606 recorded software failures, impacting half of the world's population (3.7 billion people), $1.7 trillion in assets, and 314 companies. And this is just scratching the surface—there are far more software defects in the world than we will likely ever know about.



LO$$E$ FROM SOFTWARE FAILURES (USD)

1,715,430,778,504

ONETRILLIONSEVENHUNDREDFIFTEENBILLIONFOURHUNDREDTHIRTYMILLIONSEVENHUNDREDSEVENTY-EIGHTTHOUSANDFIVEHUNDREDFOUR

https://www.tricentis.com/resources/software-fail-watch-5th-edition/

FAILURES EXAMPLES

- **Medicine infusion pumps recalled for deadly flaw.** The pump delayed the infusion which could have deadly consequences.
- **Software glitch in F-35 fighter planes causes target detection problems.** Planes, when flying in formation, were incorrectly detecting targets
- **Software bug assists in bank heist.** A printer for read-outs of the transactions made was stopped and this was discovered days later, giving thieves plenty of time for their operations.
- **Frenchman sues Uber over a software bug.** The app were pushing notifications to a device, even after it was logged out and the guy’s affairs were exposed to his wife, resulting in a divorce. Ube is sued for 45 mln $.

BIG MONEY FAILURES

- **Provident Financial stocks tank after bug disclosure.** Provident Financial, a loan company, admitted that a software bug was keeping them from collecting debts and would result in a revenue loss of up to $158,256,000. In light of the announcement, Provident Financial's stock price crashed, losing a whopping $2.2 Billion in market value in one day. Bugs in the scheduling system are arranging meetings at the wrong time.
- **St. Jude Medical patches cardiac implants after discovering how easily they can be hacked.** The cardiac implants can be remotely accessed and reprogrammed to drain the battery, interrupt pacing, or shock the patient.

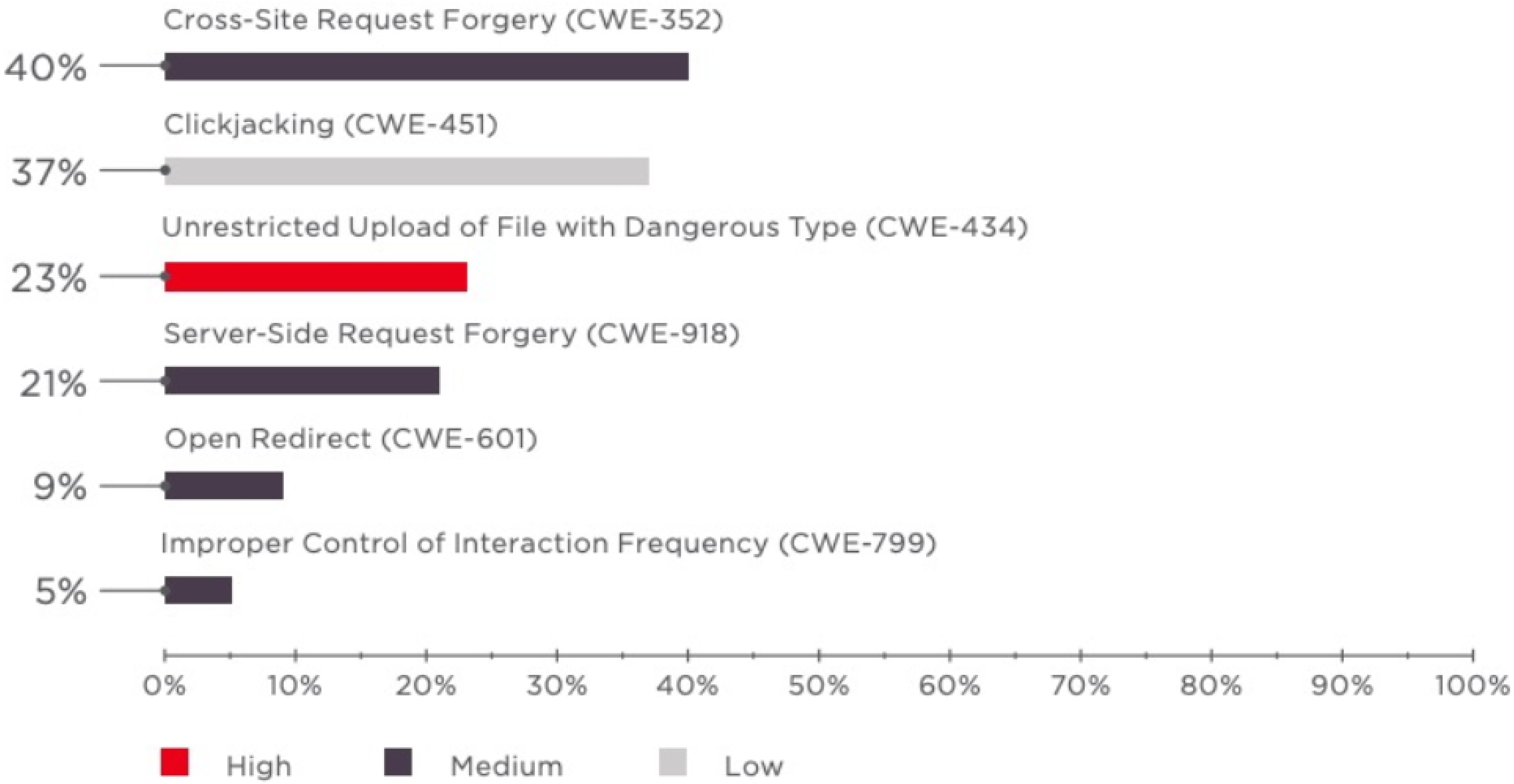
BLACK FRIDAY FAILURES

- **Walmart failure – Nov 22, 2018.** Walmart had been teasing its Black Friday deals all week, and some of the best deals were set to go live at 10pm ET, but seconds after they went live, troubles began.
- **Lowe's website crashed through Black Friday sales – Nov 23, 2018.** Lowe's apologized to customers on Twitter and temporarily shut down the site. It was relaunched on Friday afternoon
- **Neiman Marcus** - The website was down for several hours on Black Friday and the following day in 2015. It could cost them 25% of the revenue.

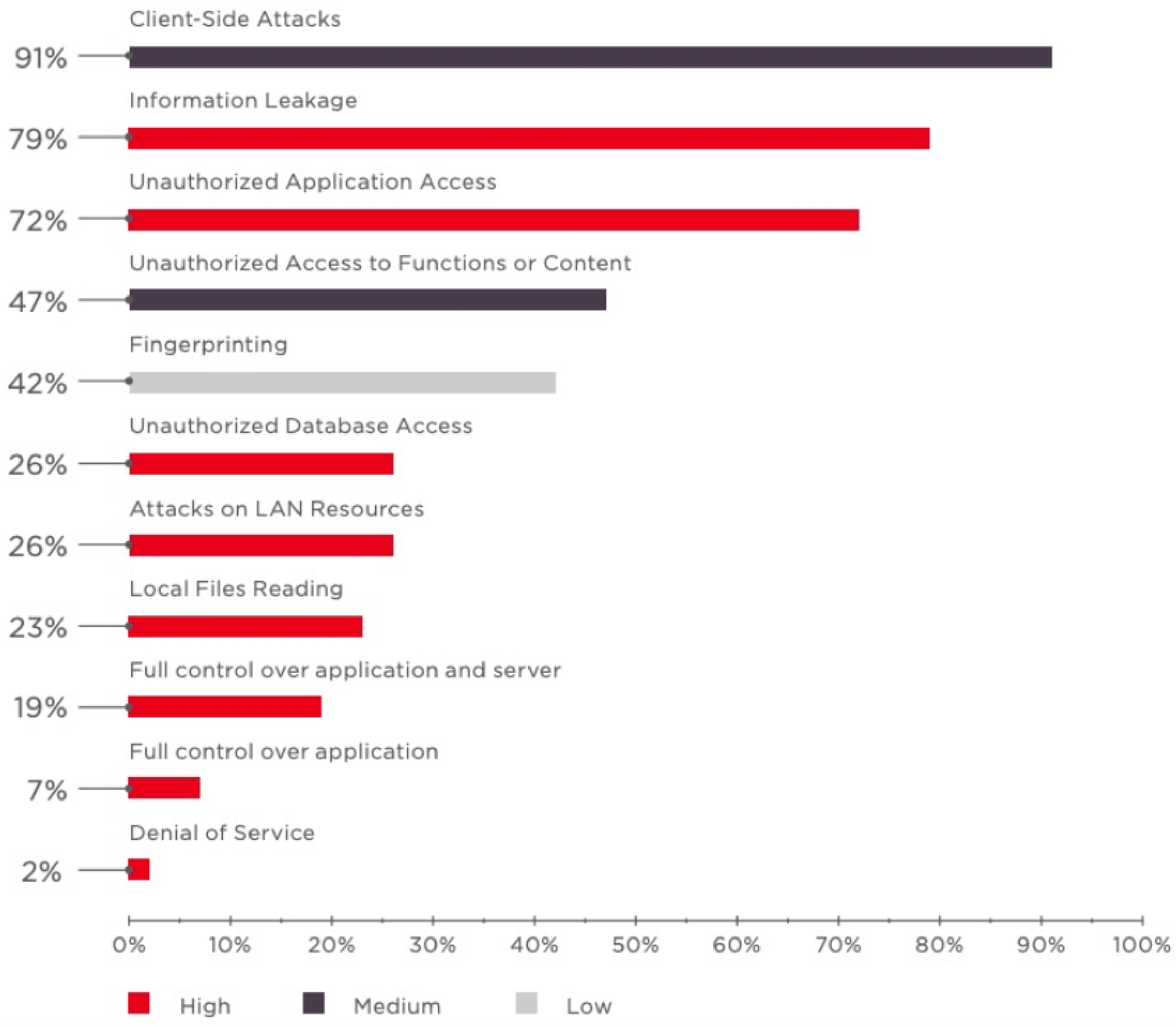
STATISTICS

- The **retail industry suffers from 2 times more SQL injection attacks than other industries.**
- Where some websites receive **4 or more attack campaigns a month,** some are constantly under attack (one of the observed websites has been under attack during 176 days out of 180 days, meaning 98% of time)
- One of the websites observed during the analysis received 94 057 SQL injection attacks in only one day (26 attacks per minute)
- The **United States** retains its rank as the **number one source of web attacks. China is the 2nd source**, but different reports tell the opposite, placing China as #1.
- The **average amount of days where a website is under attack is 12** (during a 180 days observation)
- **Average attack duration: 5 min.**

COMMON VULNERABILITIES



COMMON THREATS



PROJECT FAILS WHEN

- It is delivered out of schedule (time constraint)
- It is delivered out of budget (cost constraint)
- It is delivered out of scope (scope constraint)
- The product does not work as expected.

TOP REASONS OF FAILURES

1. Lack of Customer/User Involvement
2. Poor or incomplete Requirements Set
3. Scope Creep
4. No Change Control System
5. Poor Testing
6. Poor or insufficient communication

WHAT'S THE SOLUTION?

- Nobody knows the ideal solution to cover all the threats
- Quality issues should be addressed at the earlies phases of the project
- Proper analysis will uncover many requirements related to quality of the product
- Functional analysis should go along with non-functional analysis
- OWASP registered threats is a good start https://www.owasp.org
- Different stakeholders will reveal different requirements
- Proper management of requirements

REQUIREMENTS

OF REQUIREMENTS AND SYSTEM DEVELOPMENT

- The most important function …<in IT software system development>…. is the iterative **extraction** and **refinement** of the product **requirements.**
- The **hardest** single part of building a IT software system is deciding precisely “**what**” to build.
- No other part of the conceptual work is as difficult as establishing the detailed … requirements.
- No other part of the work so cripples the resulting system if done wrong.
- No other part is more difficult to rectify later.

Frederick P. Brooks
(in “The Mythical Man-Month”)

REQUIREMENTS DESCRIPTION

- how the system should behave,
- what properties or attributes the system should have,
- what constraints are applicable on the development process of the system.

- “Something required, something wanted or needed”. (Webster Dictionary)
- Something that the product must do or a quality that the product must have.

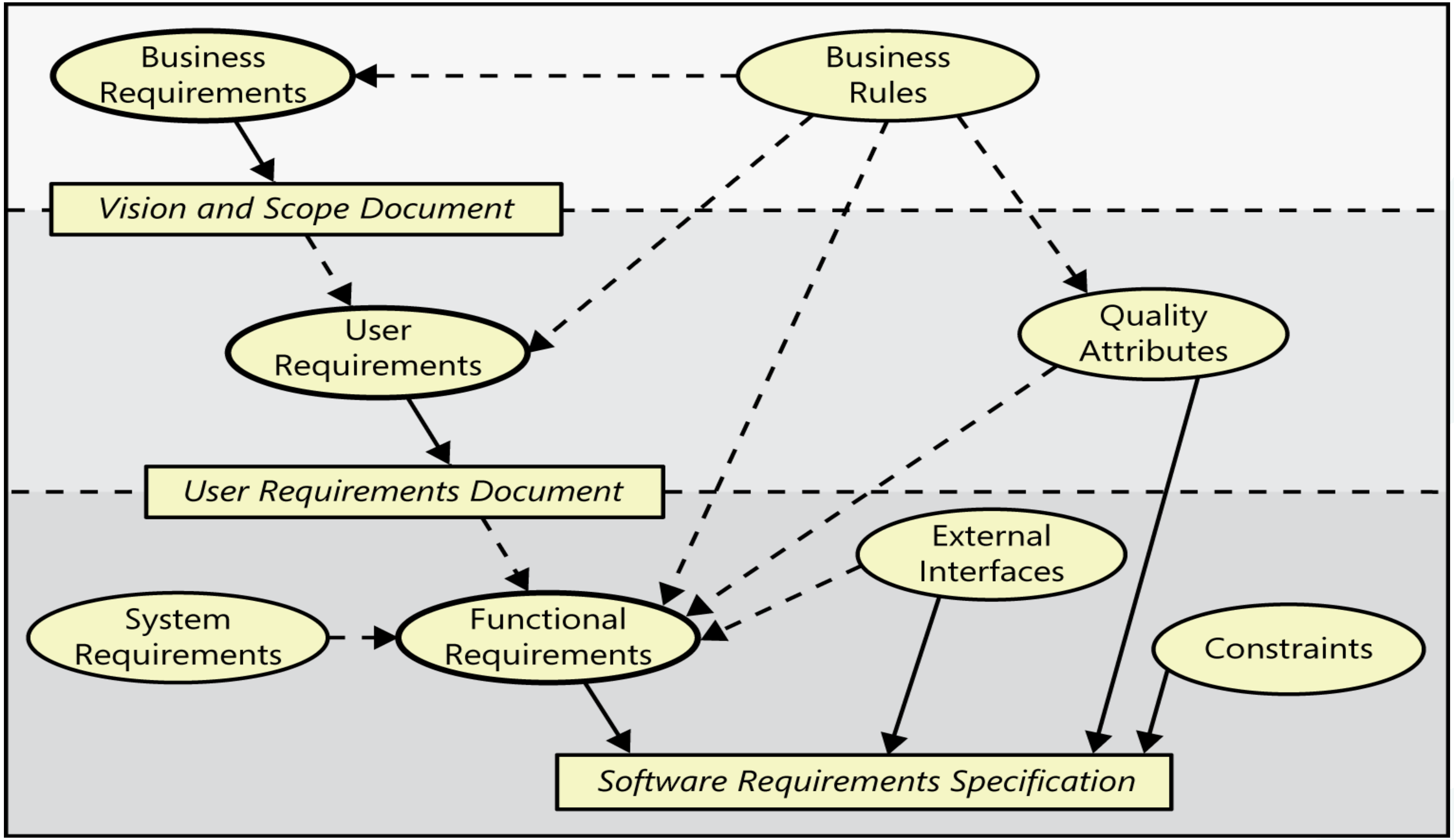
(Robertson and Robertson, 2012)

SOFTWARE REQUIREMENTS

- Software requirement is different from project requirement
- 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.
- 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.

RELATIONSHIP BETWEEN TYPES OF REQUIREMENT INFORMATION



REQUIREMENTS REPRESENTATIONS

- Requirements can be represented as documents (electronic or paper)
- More convenient is to consider requirements as containers
- Such a container can be: spreadsheets, diagrams, database, tickets.

- It is convenient to represent requirements in hierarchies of several levels
- Sometimes wireframes can take place of a part of requirements

WHAT REQUIREMENTS ARE NOT

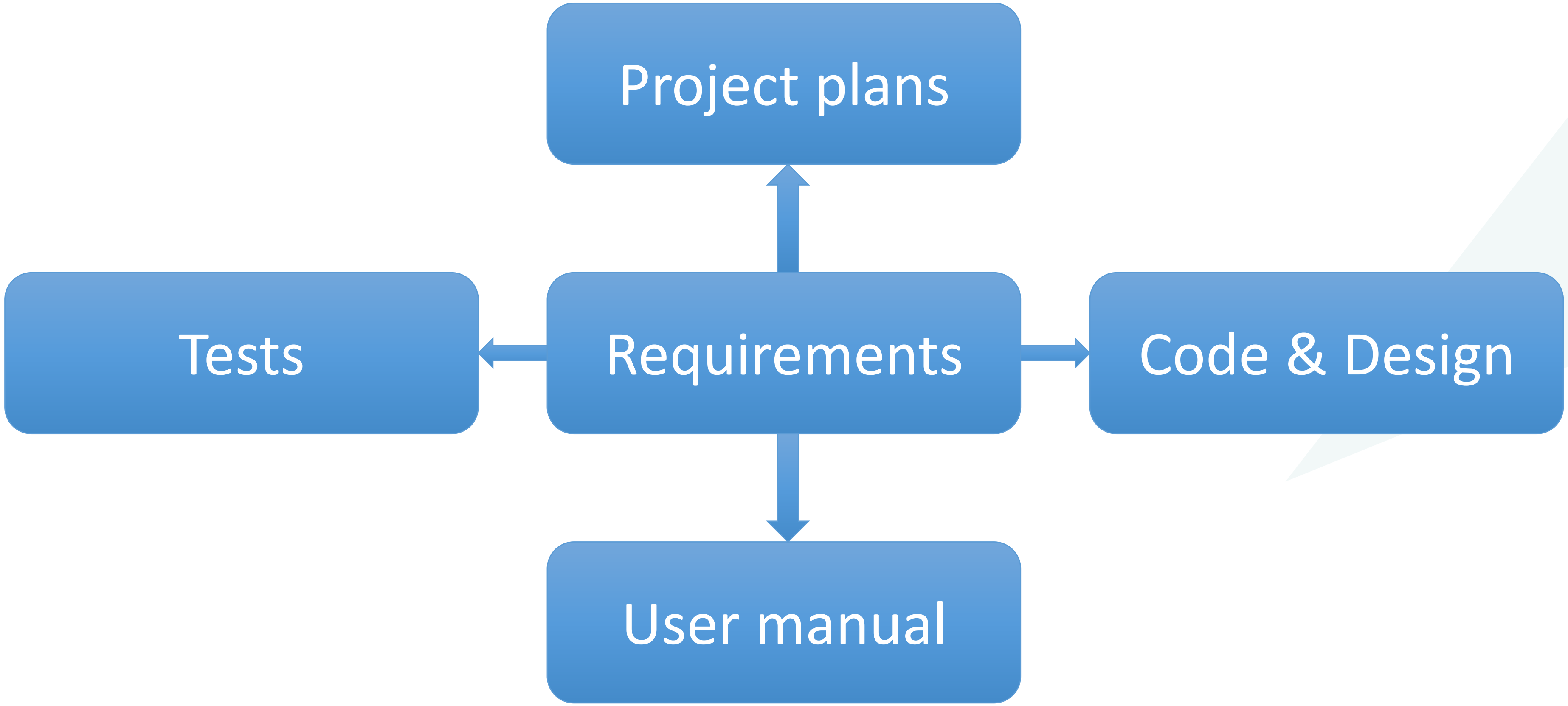
We discuss about product requirements, not project requirements

Product requirements focus on WHAT is to be built
Project requirements focus on HOW the team will build the product

Not requirements:

- Design or implementation details (other than known constraints)
- Project planning information
- Testing information
- Schedule or budget limitations
- The need for a tutorial to help new users get up to speed
- Requirements for releasing a product and moving it into the support environment

REQUIREMENTS ARE THE BASE FOR



Project plans

Tests

Requirements

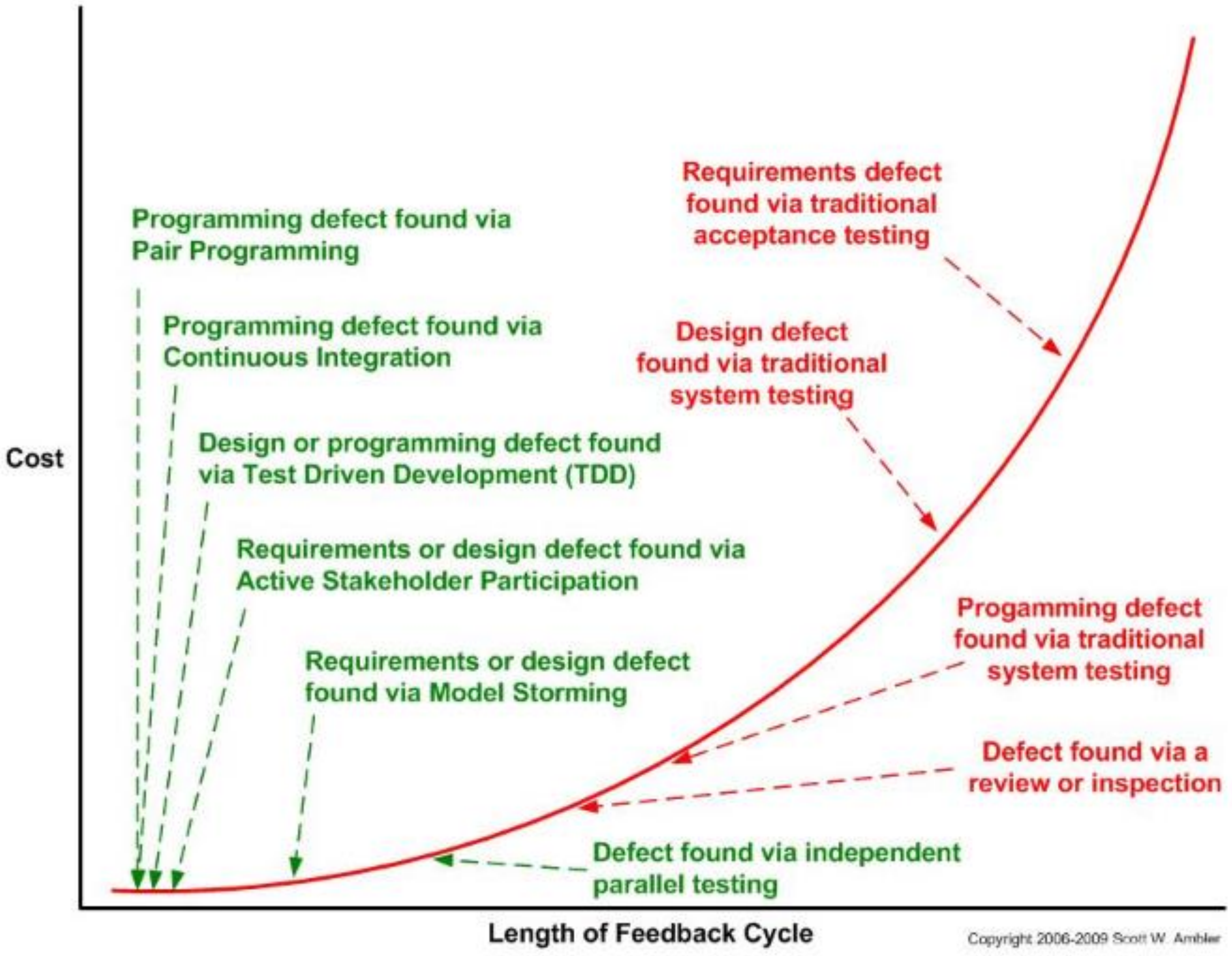
Code & Design

User manual

BENEFITS FROM A HIGH QUALITY REQUIREMENTS ENGINEERING

- Fewer defects in requirements and in the delivered product.
- 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.

COST OF DEFECT REMOVAL

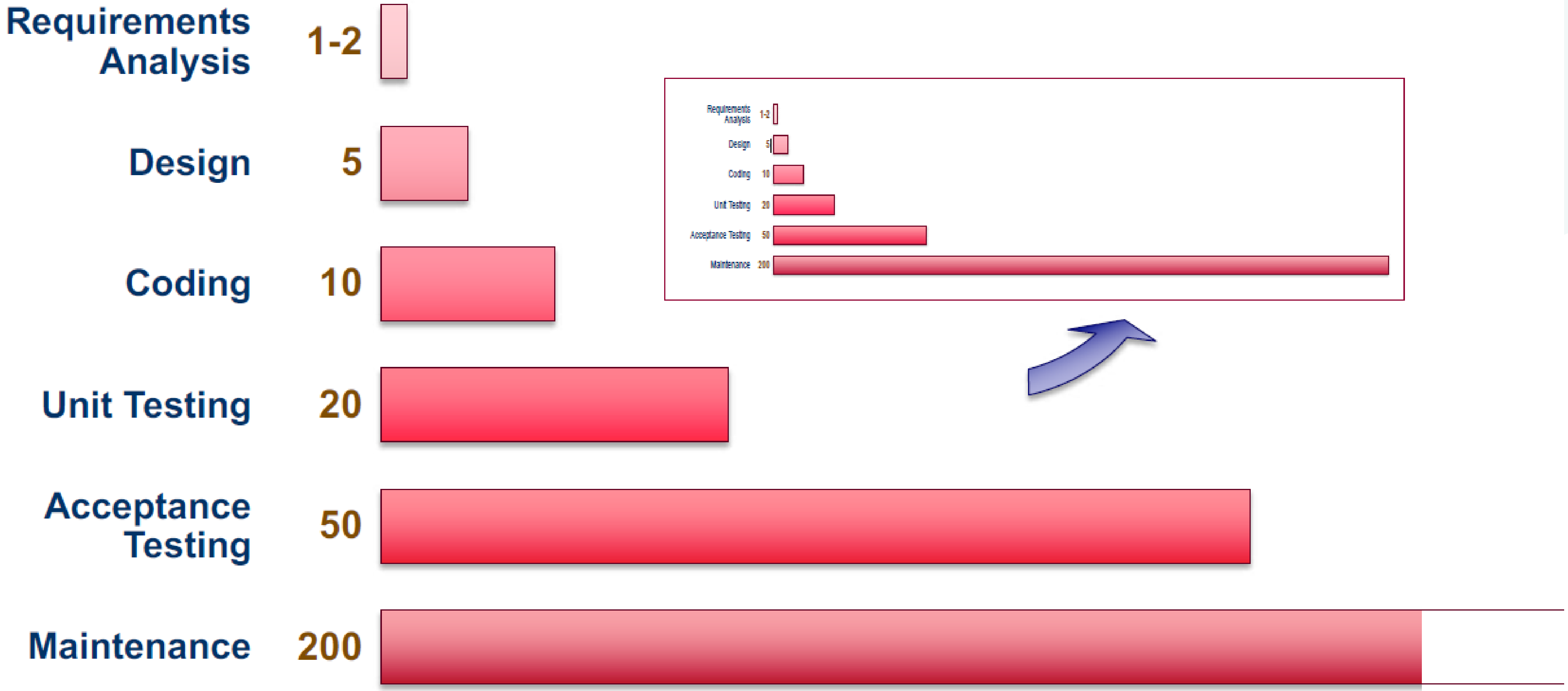


Copyright 2006-2009 Scott W. Ambler

- Get early feedback
- The earlies an error is found the less expensive it is to be fixed

COST OF DEFECT REMOVAL

Typical Lifecycle Activity ***Cost Multiplier*** *due to phase in which the defect is detected*



WHO WILL BENEFIT FORM REQUIREMENTS

Customer

- Understand and validate requirements to check that they meet the needs
- Use requirements as a means for communicating with the development team.
- Specify requirement changes.

Project Manager

- Base their plan (estimates of schedule, effort, and resources) on the product description.

Dev Team

- Use requirements to understand what to design and build.

WHO WILL BENEFIT FORM REQUIREMENTS

Testers

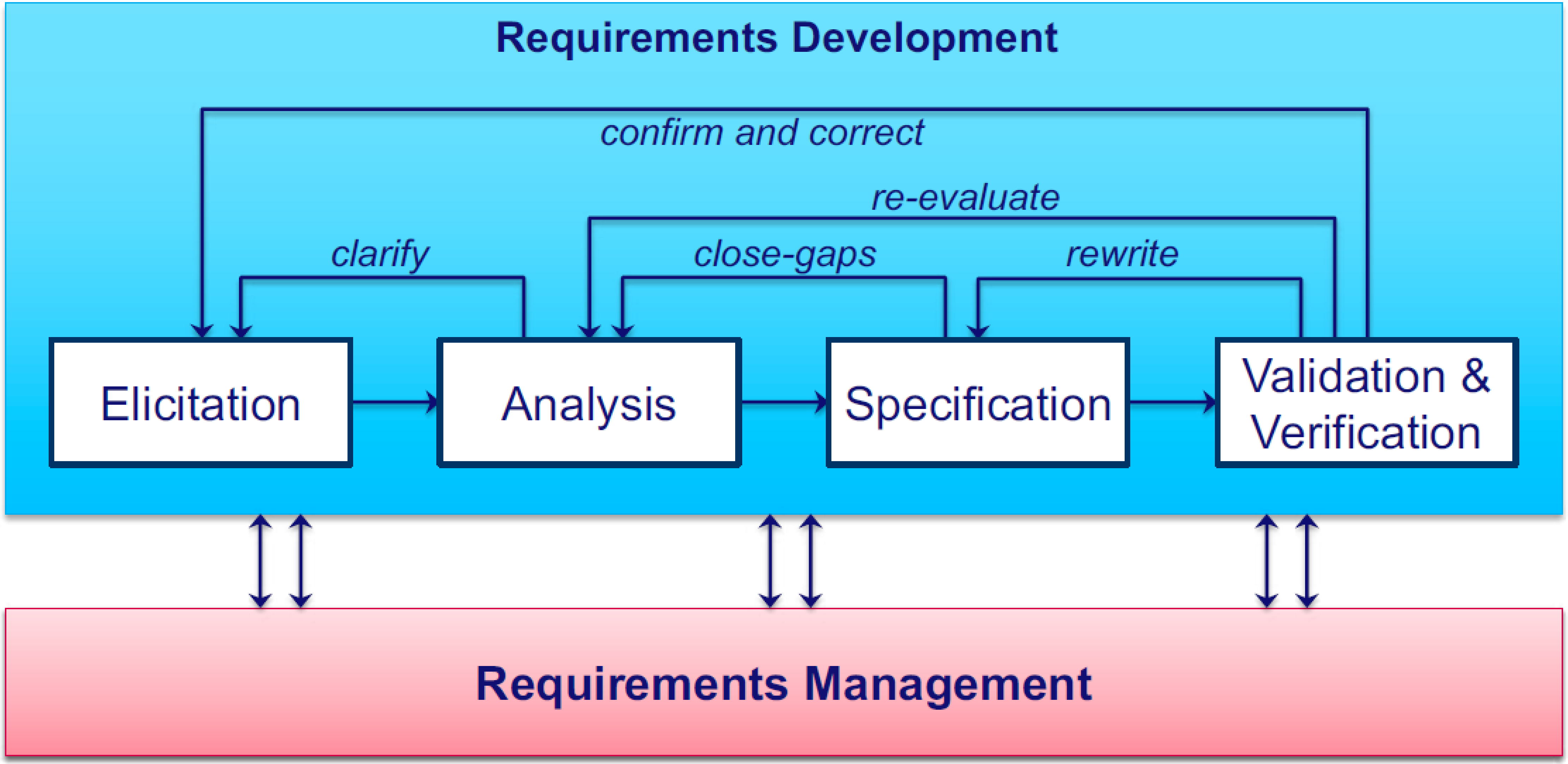
- Use requirements to develop validation tests.

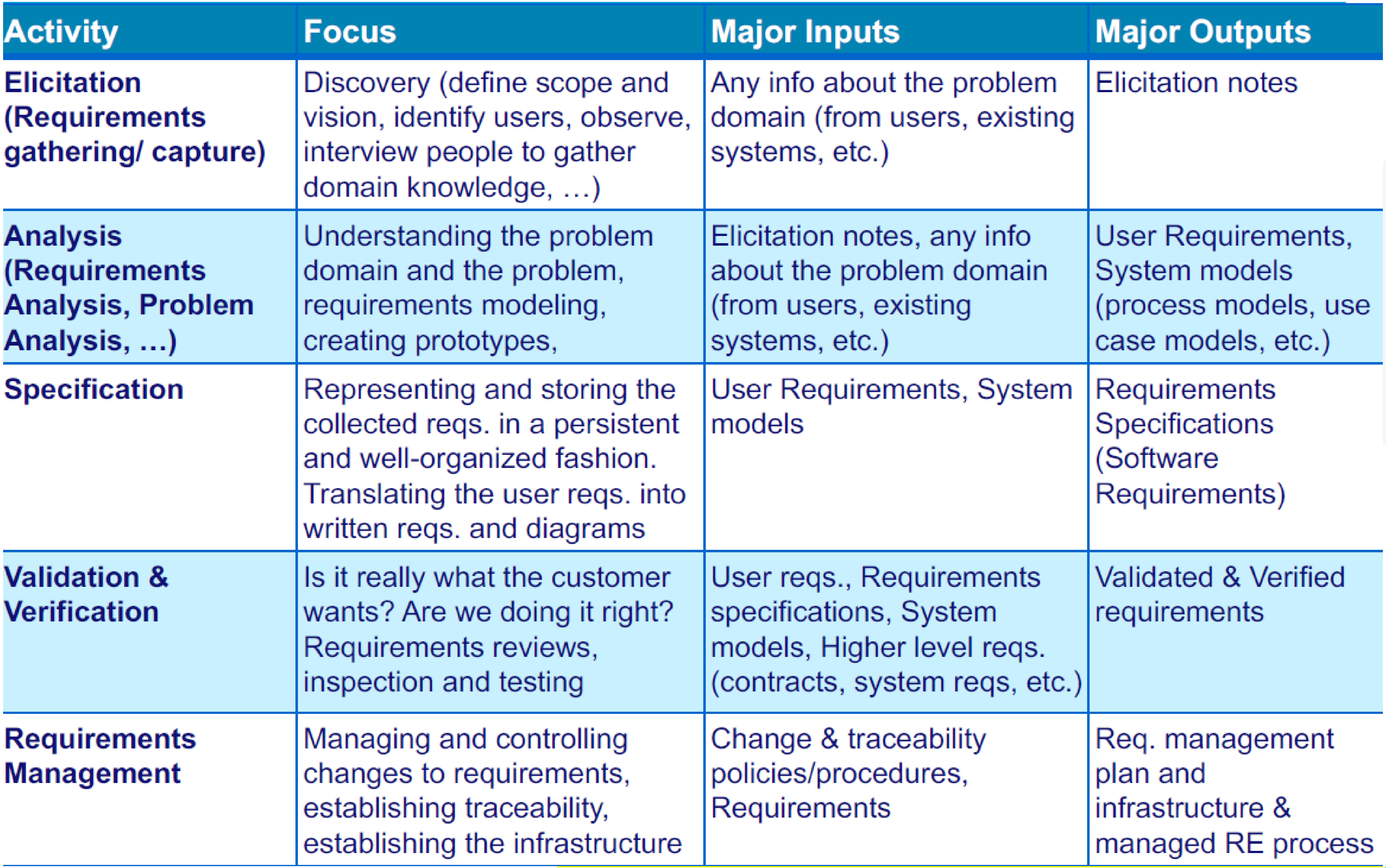
Maintenance staff

- Use the requirements to understand the system and relationships between its parts.
- Develop educational/training material for end-users, etc.

REQUIREMENTS MANAGEMENT ACTIVITIES

RE practices (activities) are interleaved, incremental, and iterative throughout requirements development & management.





Activity	Focus	Major Inputs	Major Outputs
Elicitation (Requirements gathering/ capture)	Discovery (define scope and vision, identify users, observe, interview people to gather domain knowledge, …)	Any info about the problem domain (from users, existing systems, etc.)	Elicitation notes
Analysis (Requirements Analysis, Problem Analysis, …)	Understanding the problem domain and the problem, requirements modeling, creating prototypes,	Elicitation notes, any info about the problem domain (from users, existing systems, etc.)	User Requirements, System models (process models, use case models, etc.)
Specification	Representing and storing the collected reqs. in a persistent and well-organized fashion. Translating the user reqs. into written reqs. and diagrams	User Requirements, System models	Requirements Specifications (Software Requirements)
Validation & Verification	Is it really what the customer wants? Are we doing it right? Requirements reviews, inspection and testing	User reqs., Requirements specifications, System models, Higher level reqs. (contracts, system reqs, etc.)	Validated & Verified requirements
Requirements Management	Managing and controlling changes to requirements, establishing traceability, establishing the infrastructure	Change & traceability policies/procedures, Requirements	Req. management plan and infrastructure & managed RE process

USEFUL BOOKS

Karl Wiegers, Joy Beatty: Software Requirements, 3rd Ed., Microsoft Press, 2013.

Robertson, S. and Robertson, J., Mastering the Requirements Process: Getting Requirements Right (3rd Ed), Addison-Wesley Professional, 2012

James Cadle, Malcolm Eva: Business Analysis, 3rd Ed., BCS, The Chartered Institute for IT, 2014

Alistair Cockburn: Writing Effective Use Cases, Addison-Wesley Professional, 2000

THANK YOU

AMDARIS