CHAPTER 1

Understand Software Requirement

By Aj. Dr. Pattama Longani
Collage of arts, media and Technology

Basics of Requirements Engineering

- What are the requirements?
- What are the types of requirements?
- Why do we care about requirements?
- What is the requirement engineer?
- What are qualities of good requirements

What are the requirements?

What are Requirements?

A requirement is **something the product must do** or **a quality it must have** in order to accomplish the goals of the users or the organization.

- Functional requirements: what the product must do
- Nonfunctional requirements: how well the product does things or qualities your product must have
- Constraints: restrictions on the project and the product

It is about **needs and problems**, not about solutions!

Functional Requirements

Functionality of the product (what the product must do)

- To support the business area or domain under study
- To support the product boundaries and the product's connections to adjacent systems
- To support data requirements

Nonfunctional Requirements

Product's Qualities (may not be explicitly required, but can be crucial to the success of the project)

- Look and Feel Requirements
- Usability and Humanity Requirements
- Performance Requirements
- Operational and Environmental Requirements
- Maintainability and Support Requirements
- Security Requirements
- Cultural and Political Requirements
- Legal Requirements

Project Constraints

Restrictions on the project and the product

- **limitations on the project**, and restrictions on the design of the product
- **Naming** Conventions and Definitions: the vocabulary of the project
- **Relevant Facts and Assumptions**: outside influences that make some difference to this product, or assumptions that the developers are making

What are the types of requirements?

Who are stakeholders related to a software project?

Types of Requirements

1. **Business requirements**
   - Typically come from funding sponsor for a project, the customer, marketing department, or product visionary
   - Describe why the organization is implementing the system
2. **User requirements**
   - Describe what the user will be able to do with the system.

Why do we care about Requirements?

Why do we care about Requirements?

"The hardest single part of building a software system is **deciding what to build**.

- No part of the work so cripples the resulting system if done wrong.
- No other part is more difficult to rectify later." -

Fred Brooks

Deciding what to build

- **what** the product has to do
- **how** it will be used, by **whom** it will be used
- how it **fits into** the larger picture of the **organization**, etc.
- Which **constraints** it must satisfy ?

No matter what

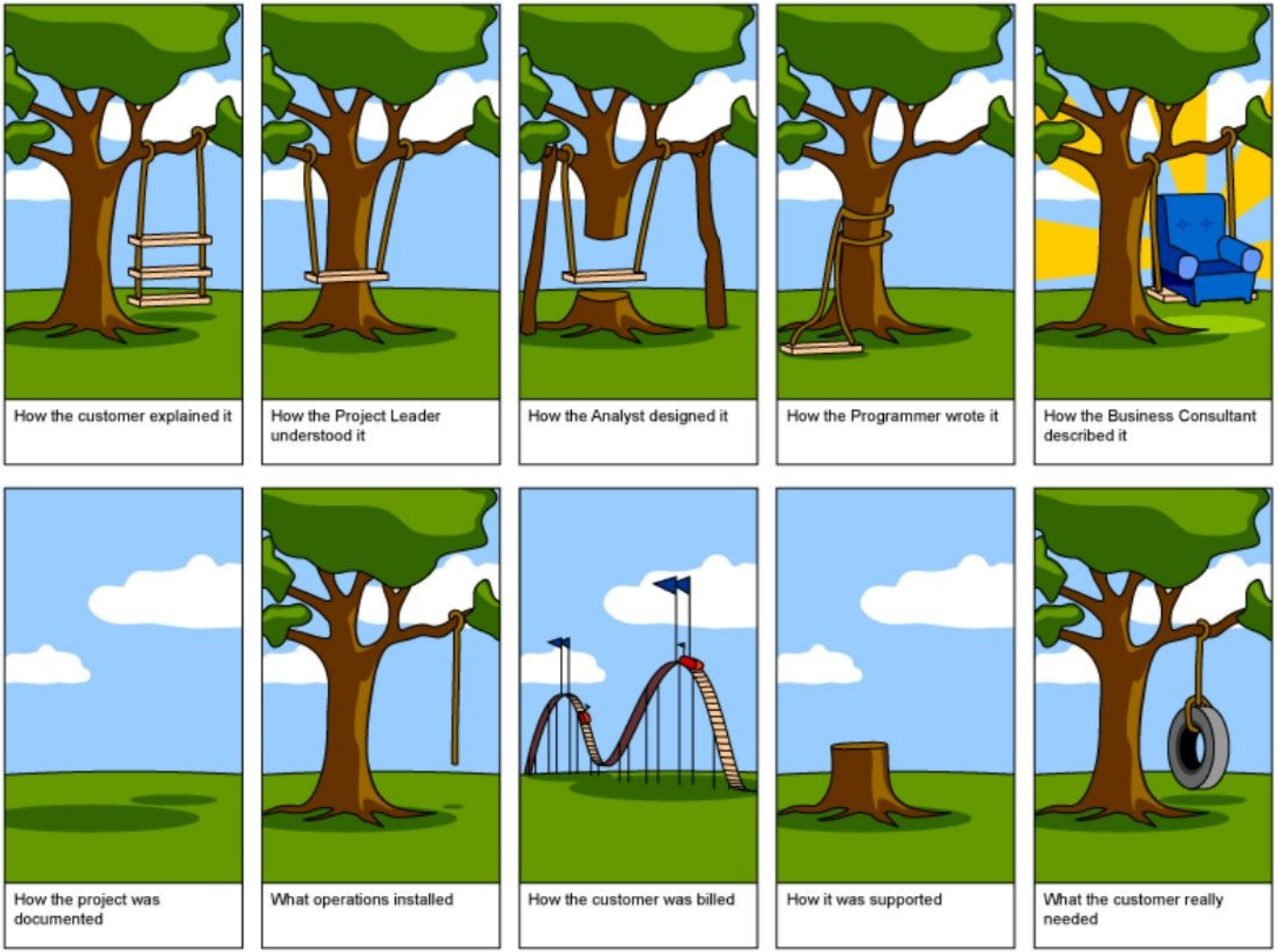
- the programming language ,
- the development tools , or
- the development process

will be used to built the right product !

The most useful **products** are those where **the developers** have understood **what** the product is intended to accomplish for **its users** and **how** it must accomplish that purpose.

You must come to the correct understanding requirement and have your client understand them.

challenges in understanding requirements



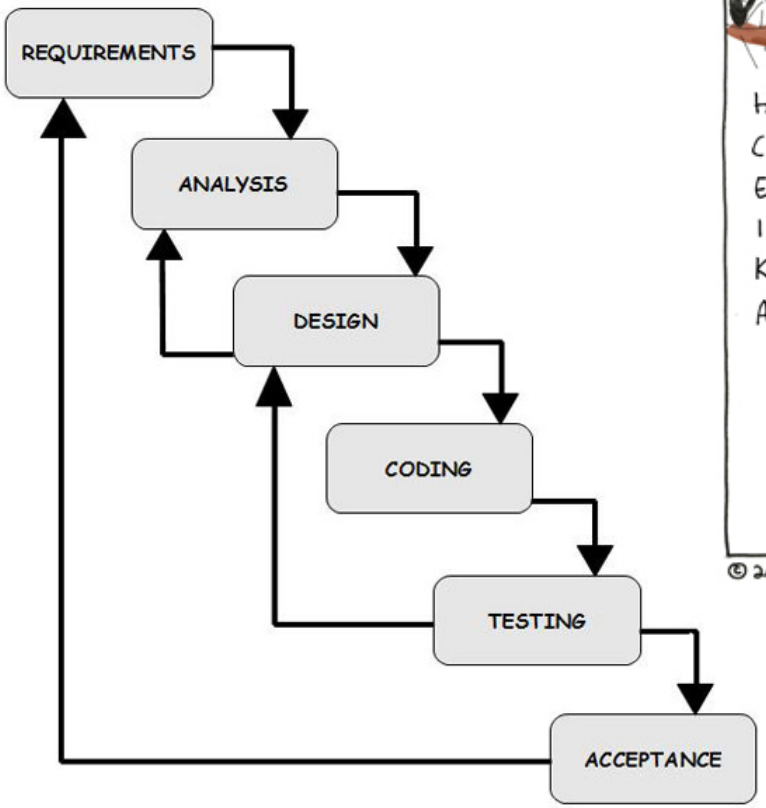
Challenges in Requirements Process

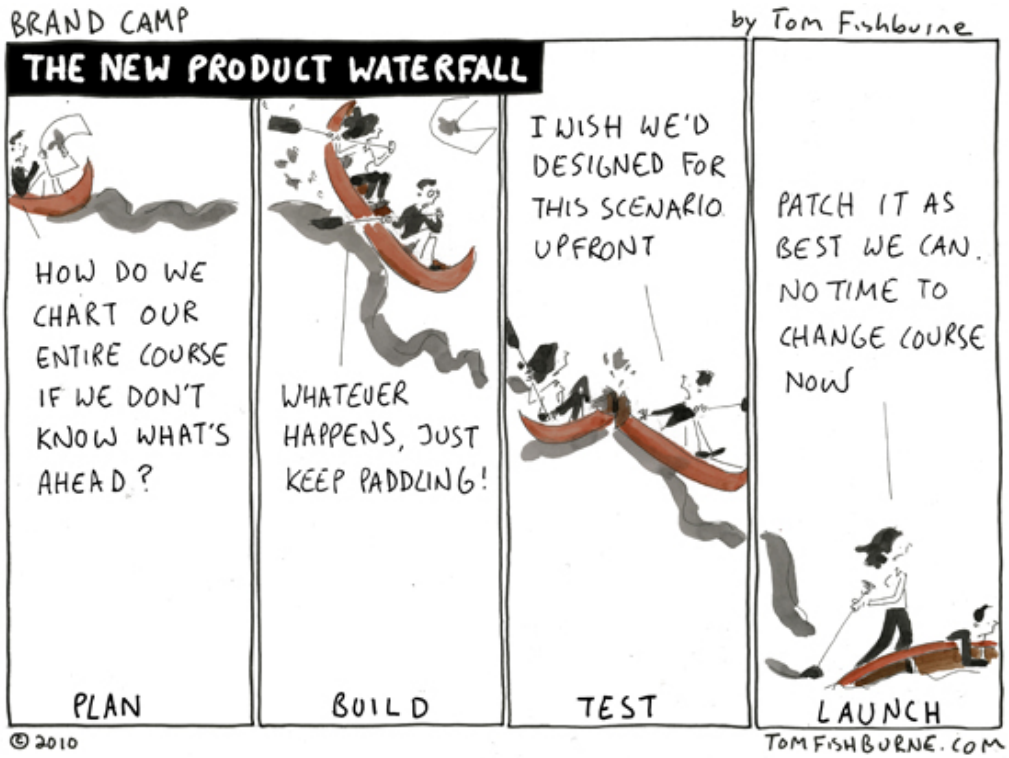
Requirements process is hard…

- Problems do not have clear boundaries.
- Success of the project is hard to measure.
- People may not know what they really want or do not speak up.
- People want different things.
- Even worse, people change their minds!
- Solutions to analysis problems are artificial.
- Eliciting and analyzing requirements need knowledge and skills.

http://www.appnovation.com/requirement_gathering_process_challenges_how_overcome_them
http://www.appnovation.com/requirement_gathering_process_challenges_how_overcome_them

Waterfall Model

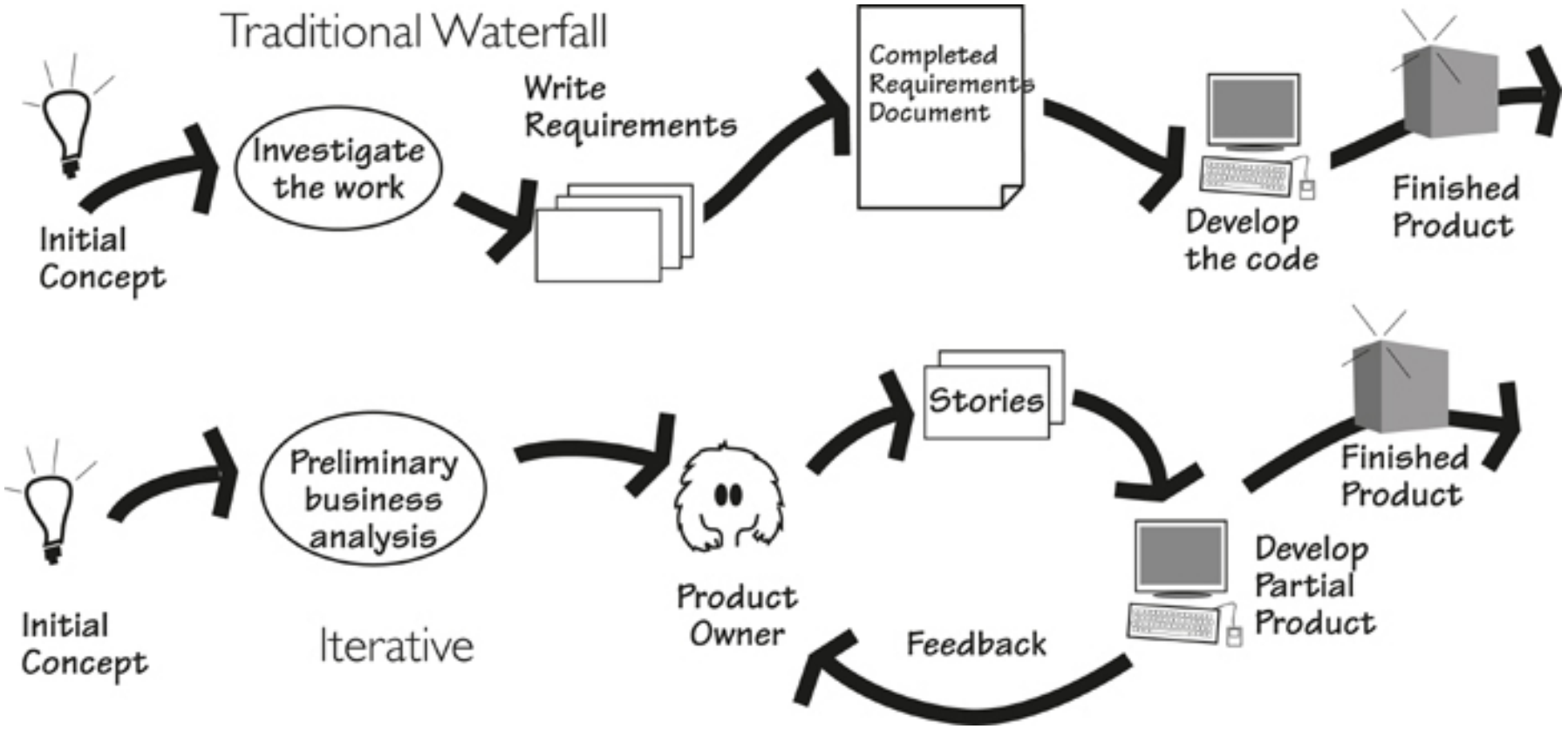




http://tomfishburne.com/2010/04/the-new-product-waterfall.html

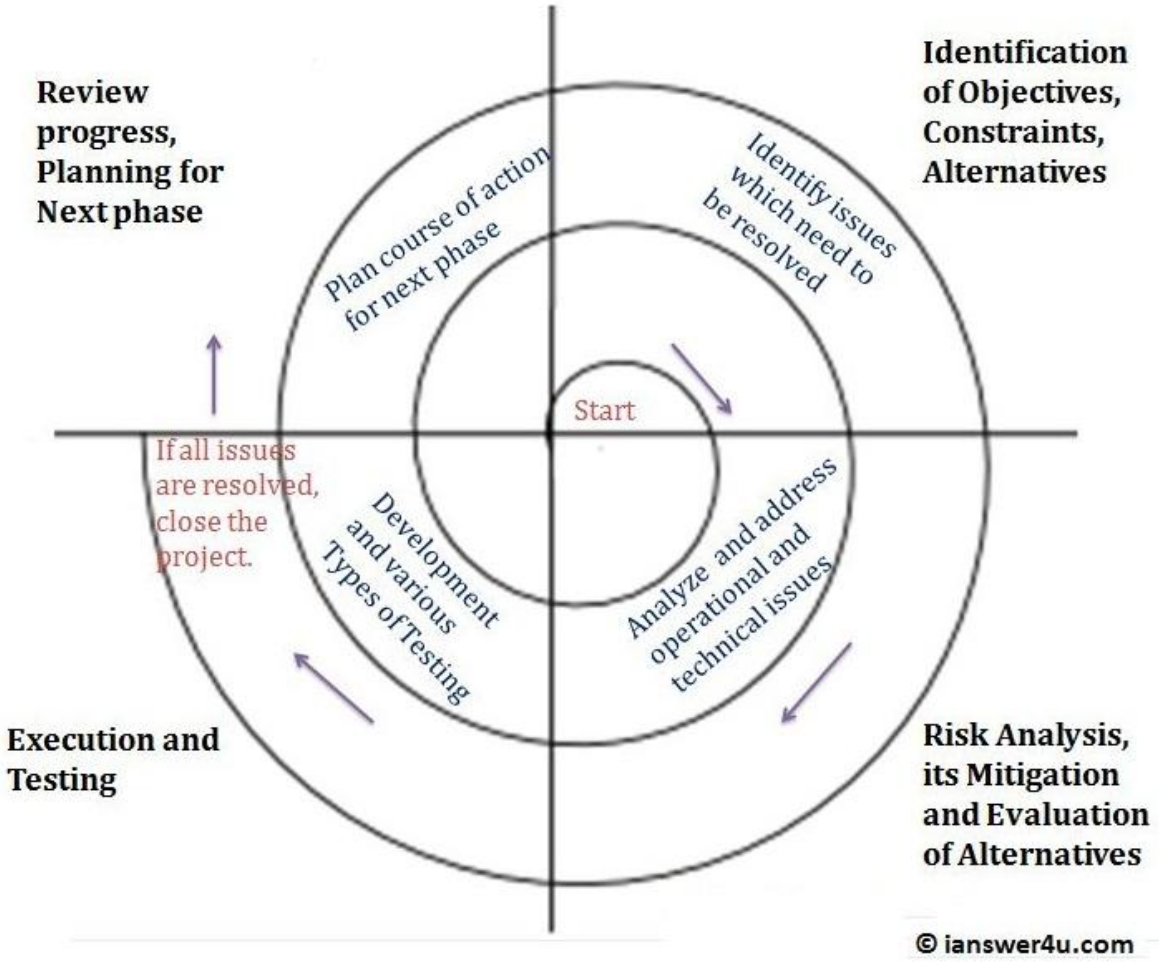
http://www.buzzle.com/articles/comparison-between-waterfall-model-and-spiral-model.html

Waterfall vs. Iterative



Mastering the Requirements Process: Getting Requirements Right, 3rd Edition

Spiral Model



http://www.ianswer4u.com/2011/12/spiral-model-advantages-and.html#axzz2UeLxfHYB

What is the requirement Engineer?

What are outputs of Requirement Analysis step?

Requirements Process By Requirement Engineering

The requirements process is meant to be a guide for producing deliverables,

- a list of things that have to be done for successful requirements projects.
- The **output** of the Requirements Process is a **written description of requirements** (**Requirement Specification**) to be used as input to the design and/or implementation of the product.

- We should learn:
  - How to **come to an understanding** of the requirements
  - How to **write them down** so that all stakeholders understand them

Requirements Process

Step	Activity	Textbook
Inception	Blastoff Meeting	Ch 3
Elicitation	Project Scoping, Identifying Stakeholders, Feasibility Analysis, Traditional approaches	Ch 3, 4
Elaboration	Trawling for Requirements	Ch 4, 5, 6
Negotiation	Functional & Nonfunctional Requirements, Constraints, Cost & Risks, Fit Criteria, Methodologies and Techniques	Ch 7, 8, 9, 12
Specification	Writing Requirements	Ch 10
Validation	Quality Gateway	Ch 11
Requirement Management	Reusing Requirements, Reviewing Specification Managing Requirements for later use	Ch 13, 14, 15

What are qualities of good requirements?

Qualities of a Good Requirement

- "I need to ensure that each requirement is **unambiguous**; that is, it must have the same meeting to both the client and the developer. I also need to **measure** the requirement against the client's expectations. If I can't put a measurement to it, then I can never tell if we are building the product the client really needs." (Textbook, p29)

Qualities of a Good Requirement

- Easy to understand
- Concise (1 sentence, 1 verb)
- Unambiguous
- Consistent (names, vocabularies)
- Testable
- About what, not about how

For A Good Specification

Preventing requirements leakage through the **Quality Gateway**

- Any missing requirements?
- Are requirements consistent?
- Any potential conflicts between requirements?
- Does every part have the same understanding?
- Are requirements achievable with the given time and budget?

When Will we decide to build a program?