# Best Practices in Business Analysis

## Session 1: Mastering the Requirements Process

# Welcome to the Course

- High level overview into real world business analysis
- Based on observation of what the best BAs ***actually*** do
- Three sessions
  - *Session 1*: Mastering the Requirements Process
  - *Session 2*: Developing High Quality Specifications
  - *Session 3*: Boots on the Ground – Working with Stakeholders
- Objectives:
  - To identify the best practices and methods used by the best BAs
  - Explain why these are best practices
  - Give you a road map to adopting these practices yourselves
  - Provide a set of resources for you as a BA
- Course Goal: To give you something you can use and get results with

“No one ever gives me requirements,
I get requirements whether people
want to give them or not”

*Dmitri the BA*

“For decades we have deluded ourselves into believing that users actually know their requirements and all we have to do is ask them. ”

*Ivar Jacobson*

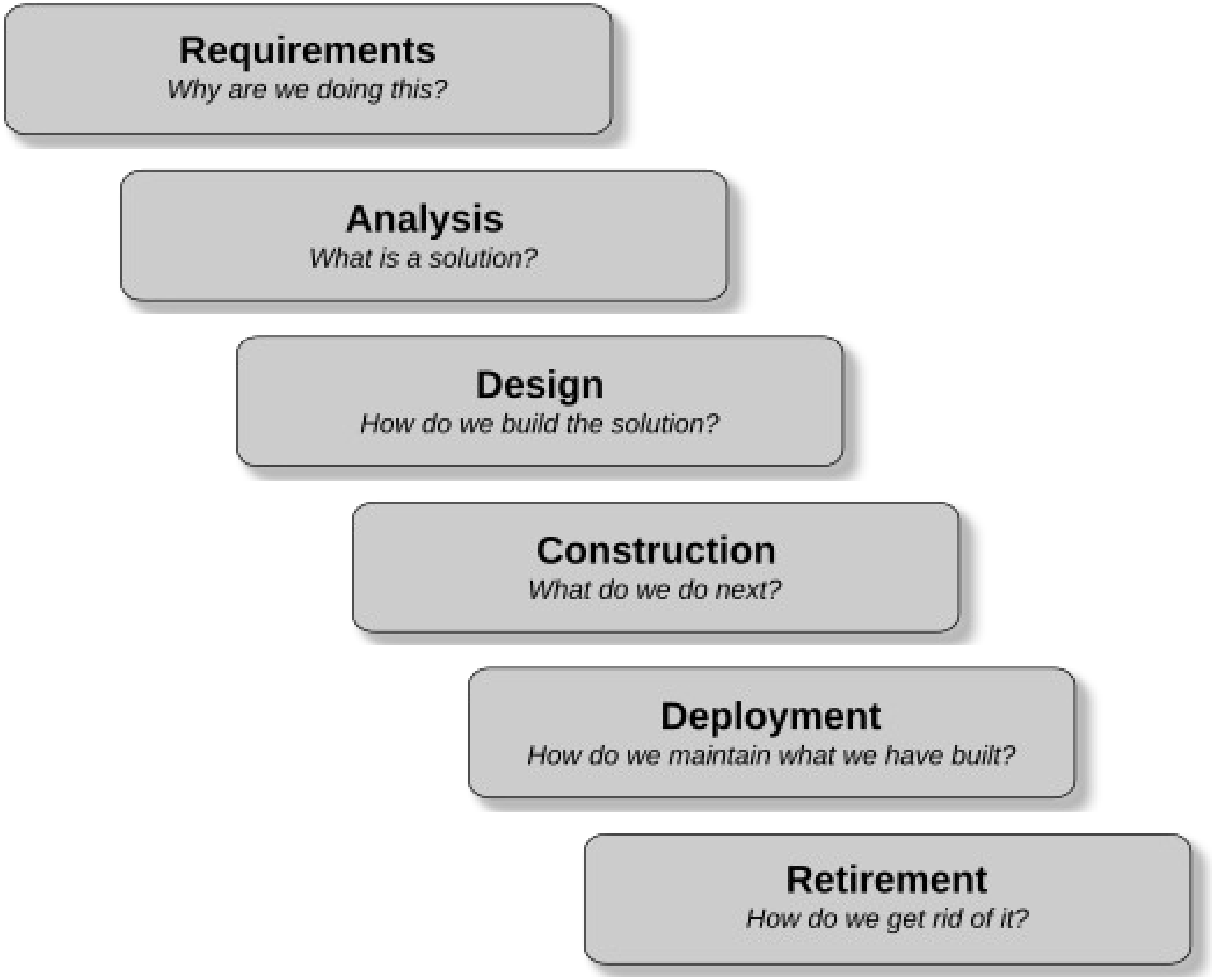
# Session Outline

- The world of the BA: roles, challenges, functions and needed skills
- The importance of high quality requirements and specifications
- The requirements process: how the experts manage requirements
- Typical problems and challenges
- Developing a collaboration framework

# The BA Role

- The BA role is not unique to software development
- The BA function exists in other “production” environments
  - Identify needs of stakeholders
  - Translate those needs into solution
- BA functions are often called different things
  - Sales people
  - Investigators
  - Negotiators
- BA Role = “problem identification and solution discovery”

# The Engineering Process



**Requirements**
*Why are we doing this?*

**Analysis**
*What is a solution?*

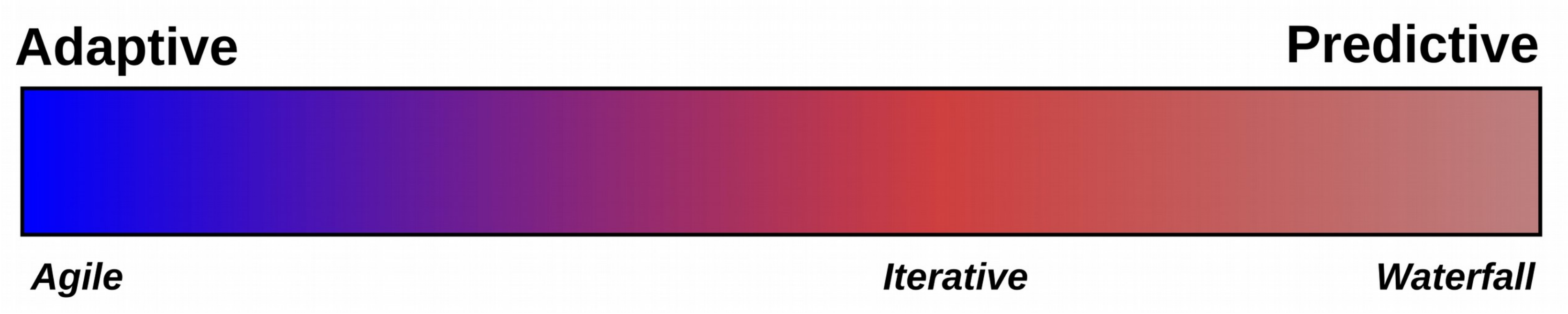
**Design**
*How do we build the solution?*

**Construction**
*What do we do next?*

**Deployment**
*How do we maintain what we have built?*

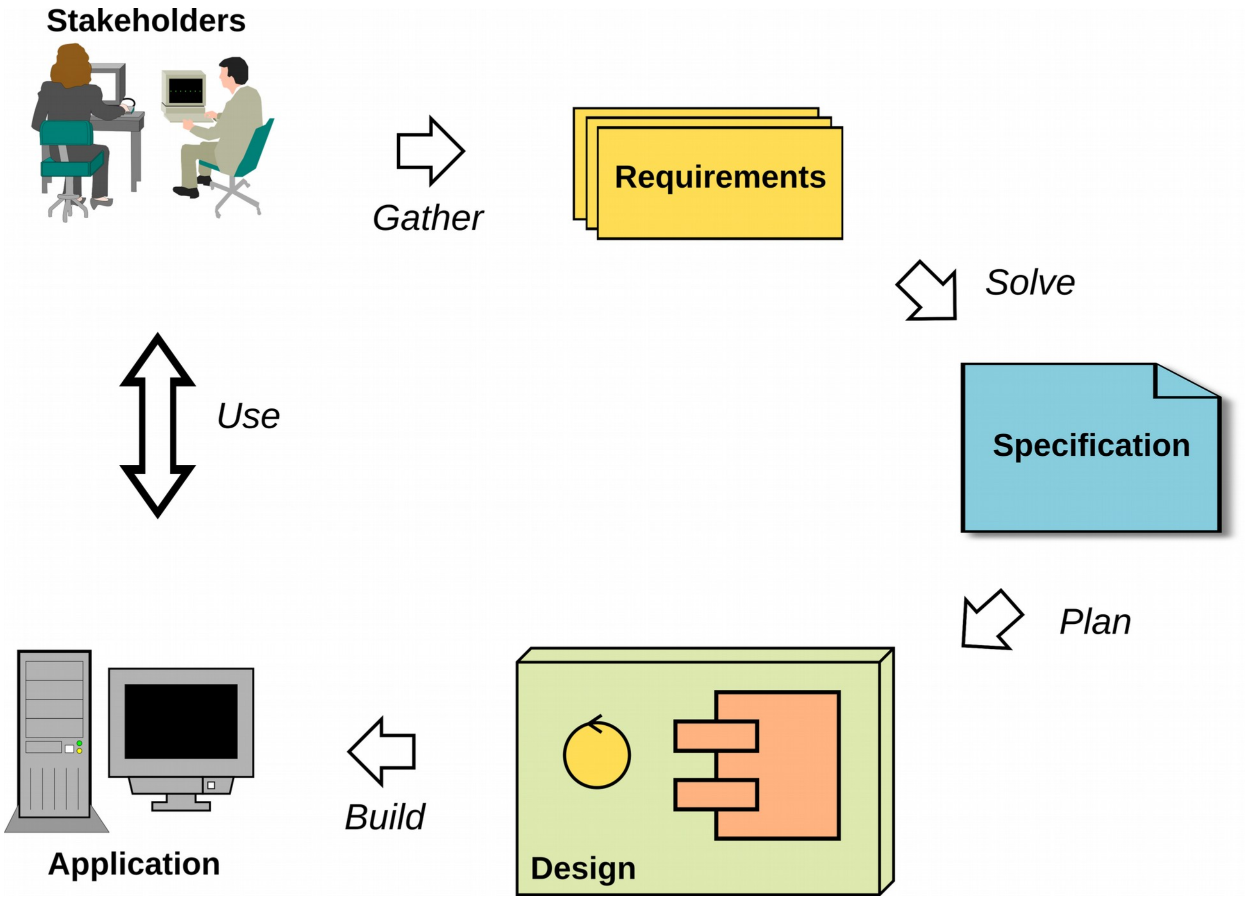
**Retirement**
*How do we get rid of it?*

# The Process Types

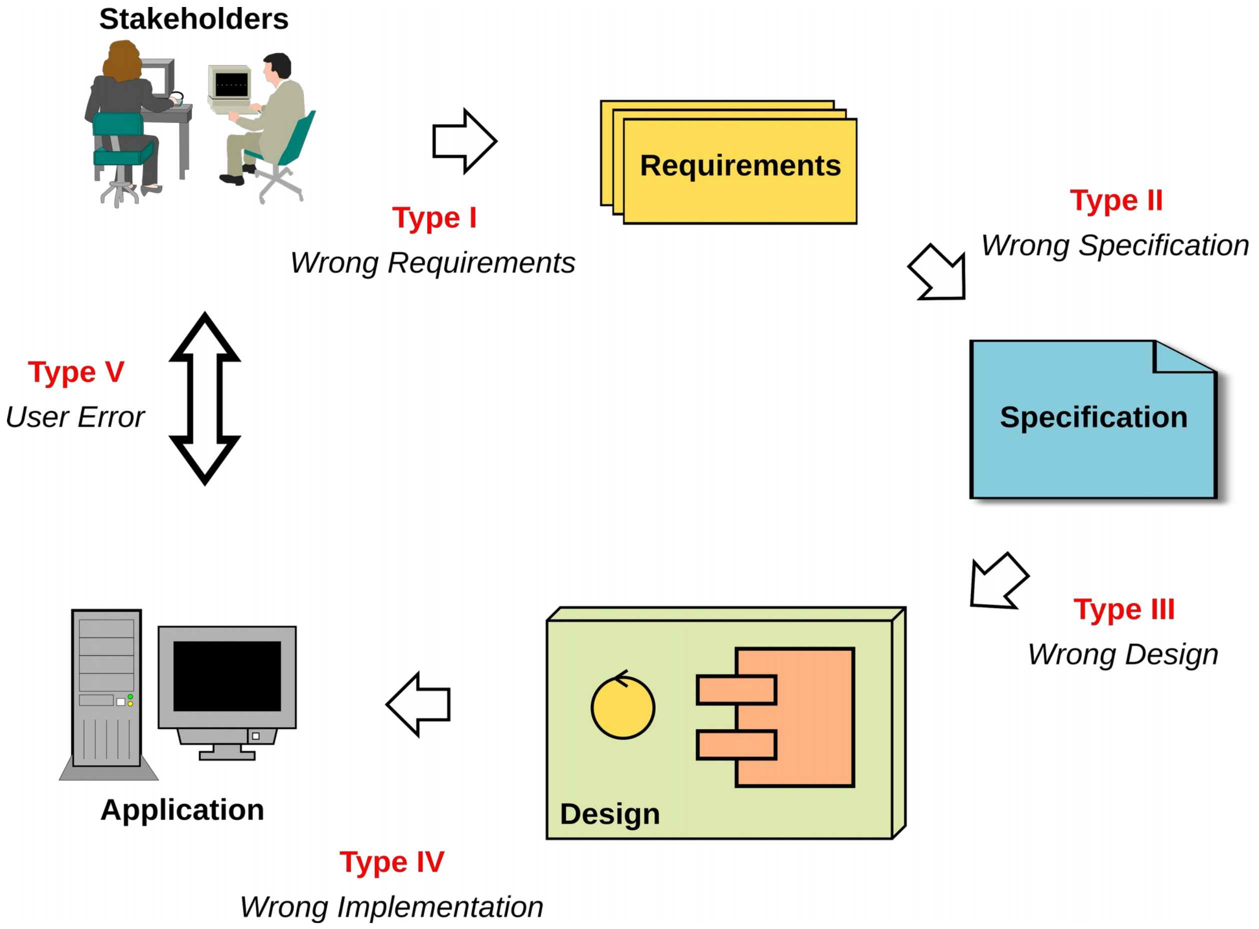


- All processes use the steps of the engineering process
- The difference is how they apply the steps

# The Generic SPD



# Where Things Go Wrong



Stakeholders

Requirements

Specification

Design

Application

**Type I**

*Wrong Requirements*

**Type II**

*Wrong Specification*

**Type III**

*Wrong Design*

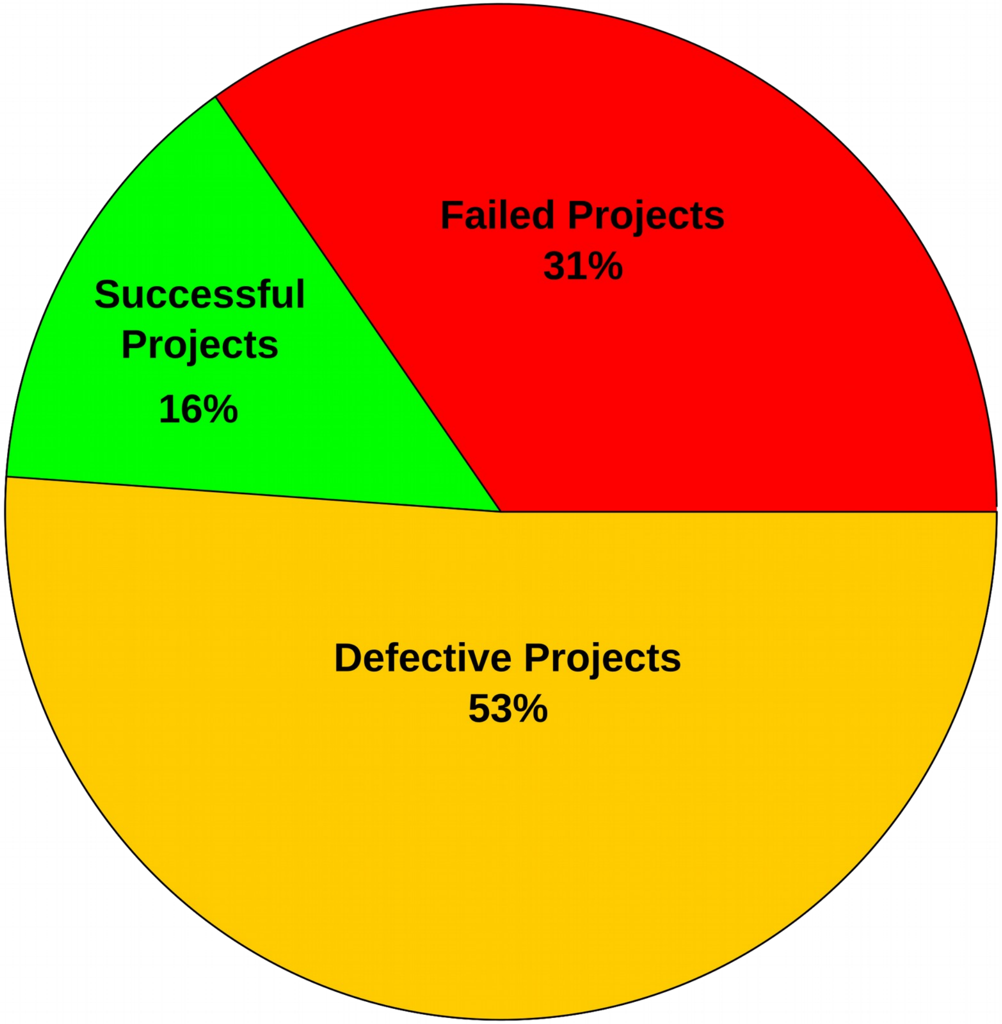
**Type IV**

*Wrong Implementation*

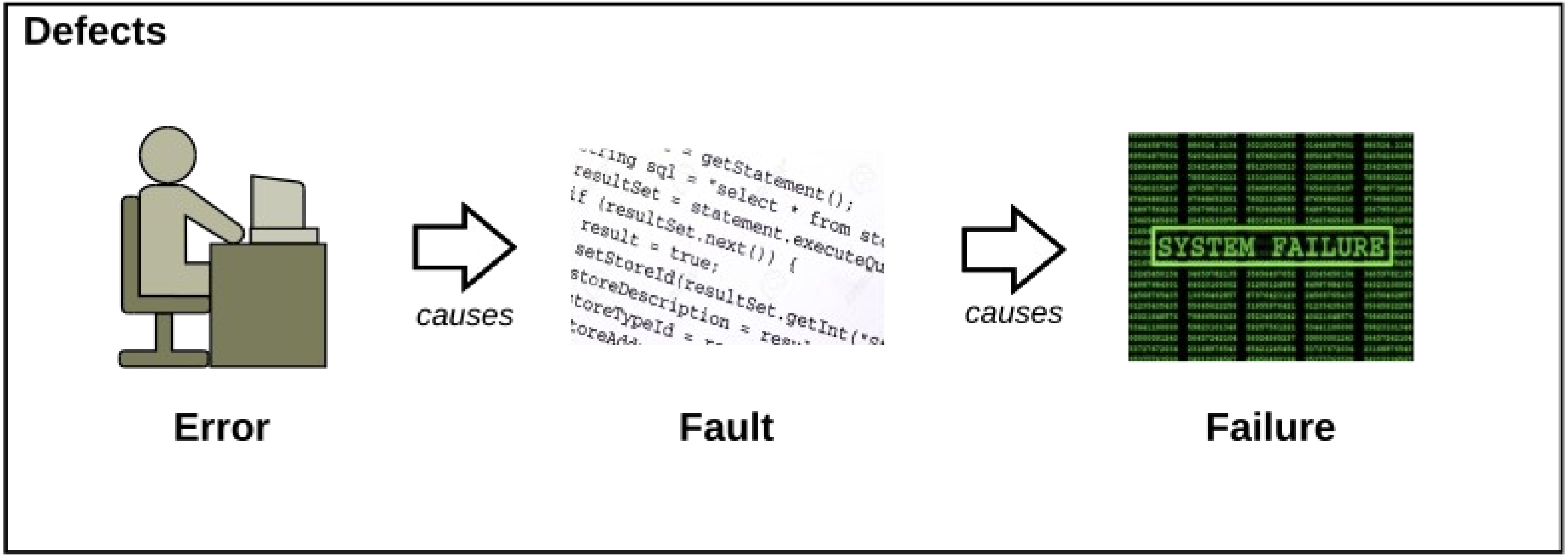
**Type V**

*User Error*

# The Chaos Report



# Root Causes of Defects



Defects

Error → causes → Fault → causes → Failure

```
...ring sql = "select * from st
resultSet = statement.executeQu
if (resultSet.next()) {
 result = true;
setStoreId(resultSet.getInt("s
storeDescription = resu
toreTypeId = re
toreAdd
```

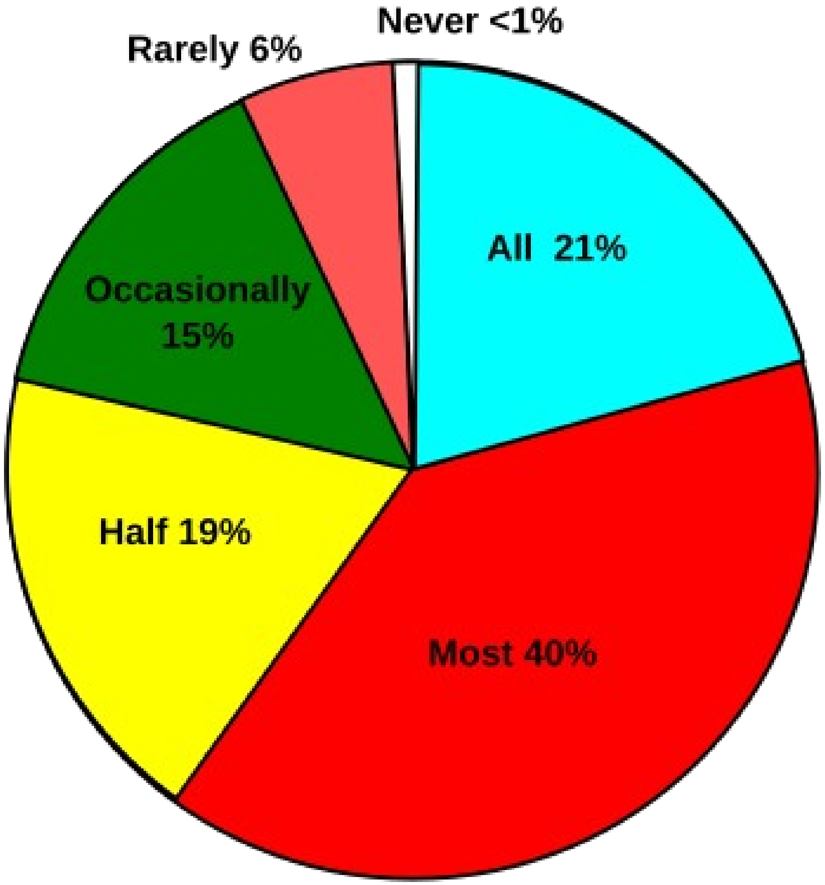
**Error:** a human action that eventually leads to a fault

**Fault:** an incorrect step in building the system at any point that results in failure

**Failure:**  any place the software does not perform as required

**Defect:** a generic term for any of the above

# Rework



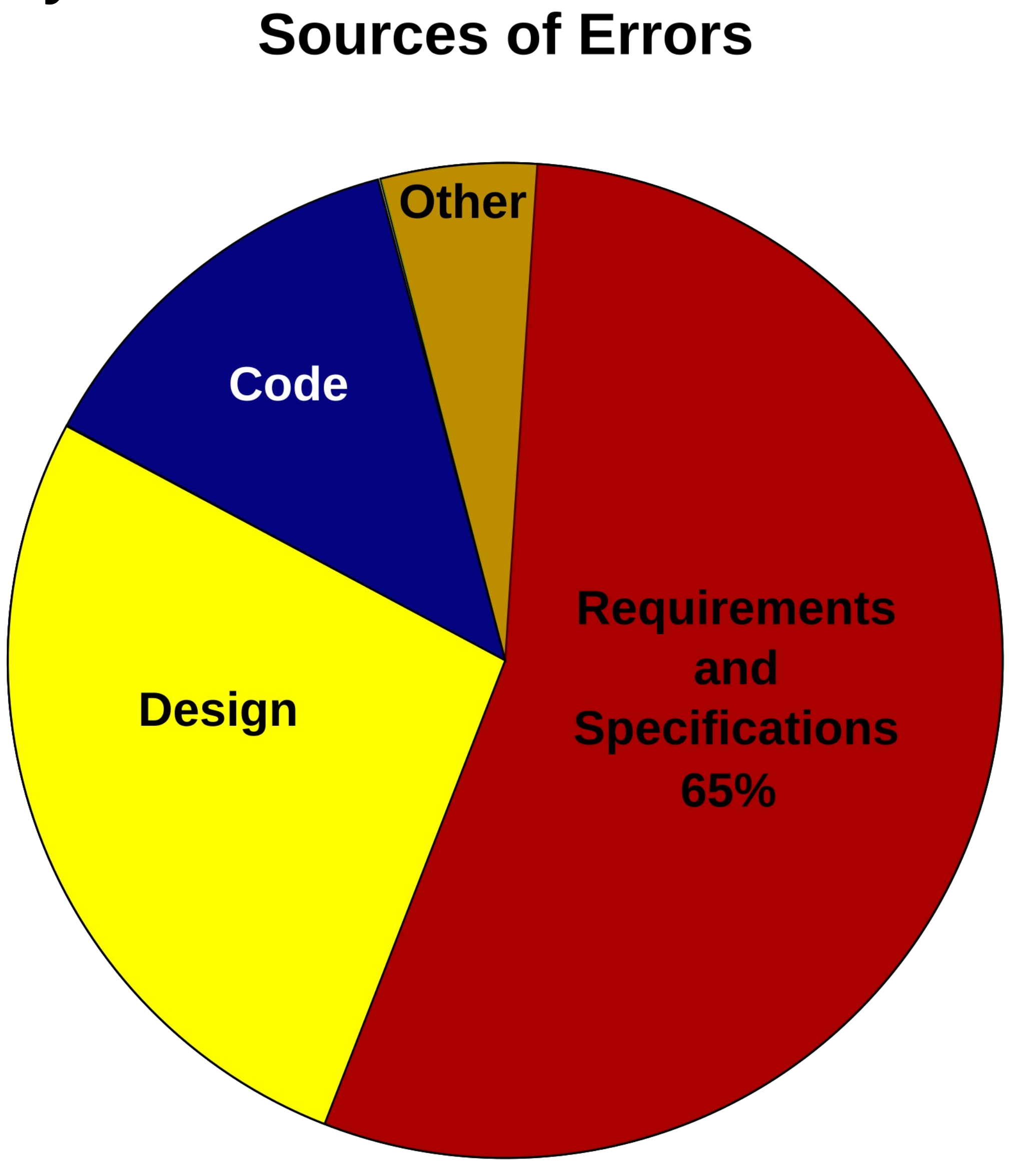
:

**Geneca Survey of IT Professionals:**

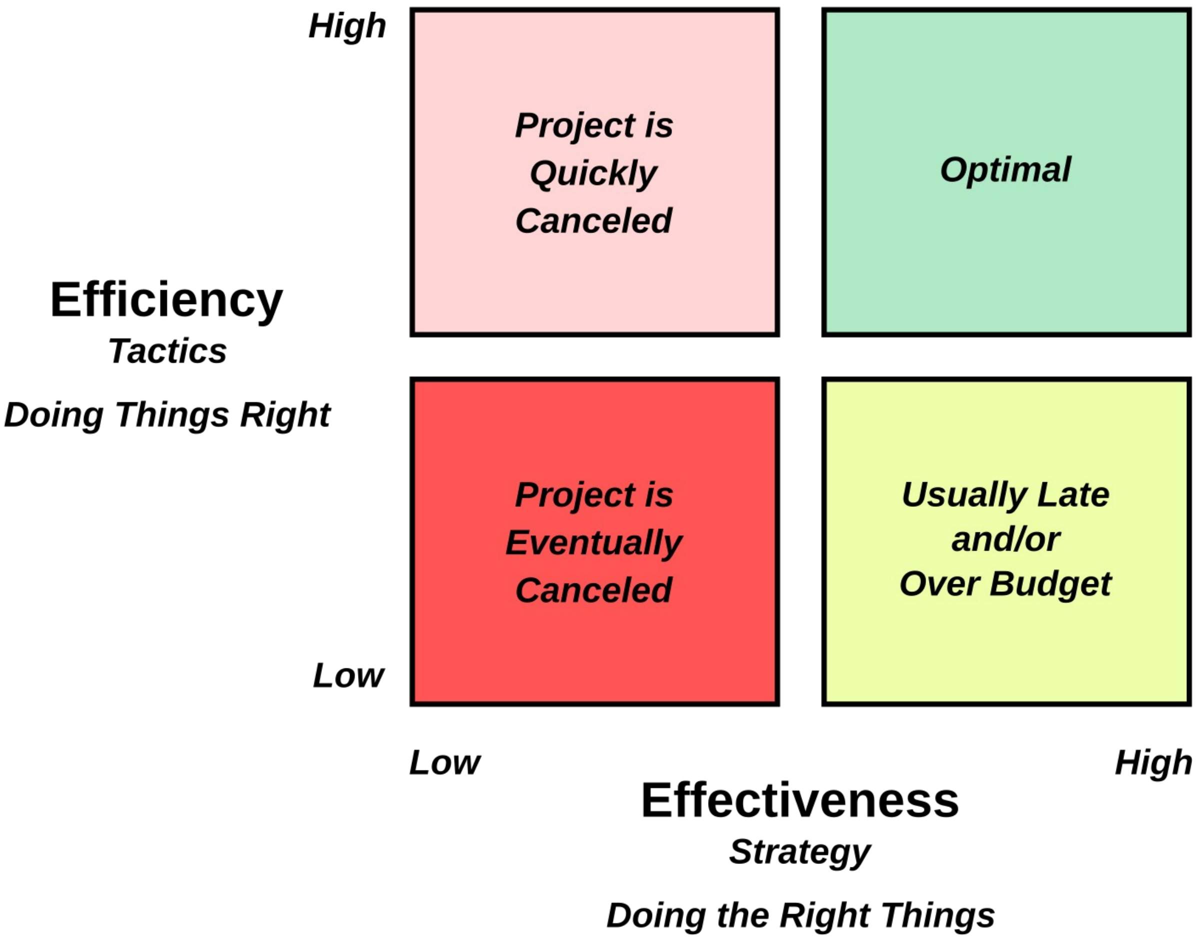
*"How much of your time do you spend on avoidable or preventable rework (eg. changing requirements, missing features)?"*

*Results are consistent with other studies.*

# Patel’s Study



# Efficient and Effective



High

Project is Quickly Canceled

Optimal

Efficiency

Tactics

Doing Things Right

Project is Eventually Canceled

Usually Late and/or Over Budget

Low

Low

High

Effectiveness

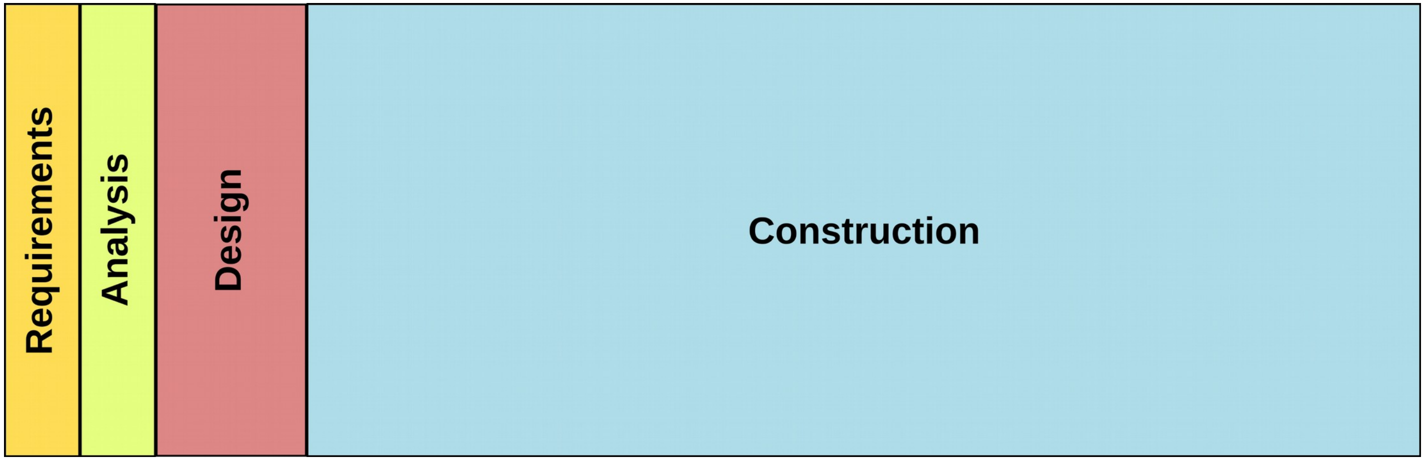
Strategy

Doing the Right Things

# The Actual Costs of Rework

*Reworking defective requirements, design, and code typically consumes 40 to 50 percent of the total cost of software development, and can be as high as 80%.*

*Construction typically consumes 80% or more of the resources of a project. The whole purpose of the Requirements, Analysis and Design phases is to reduce the risk of costly errors that force rework in the construction phase*



“[requirements gathering] is frustrating, full of complex interpersonal relationships, indefinite and difficult.

In a word, it is fascinating. Once you’re hooked, the old easy pleasures of building a system are never again enough to satisfy you.”

*Tom DeMarco*

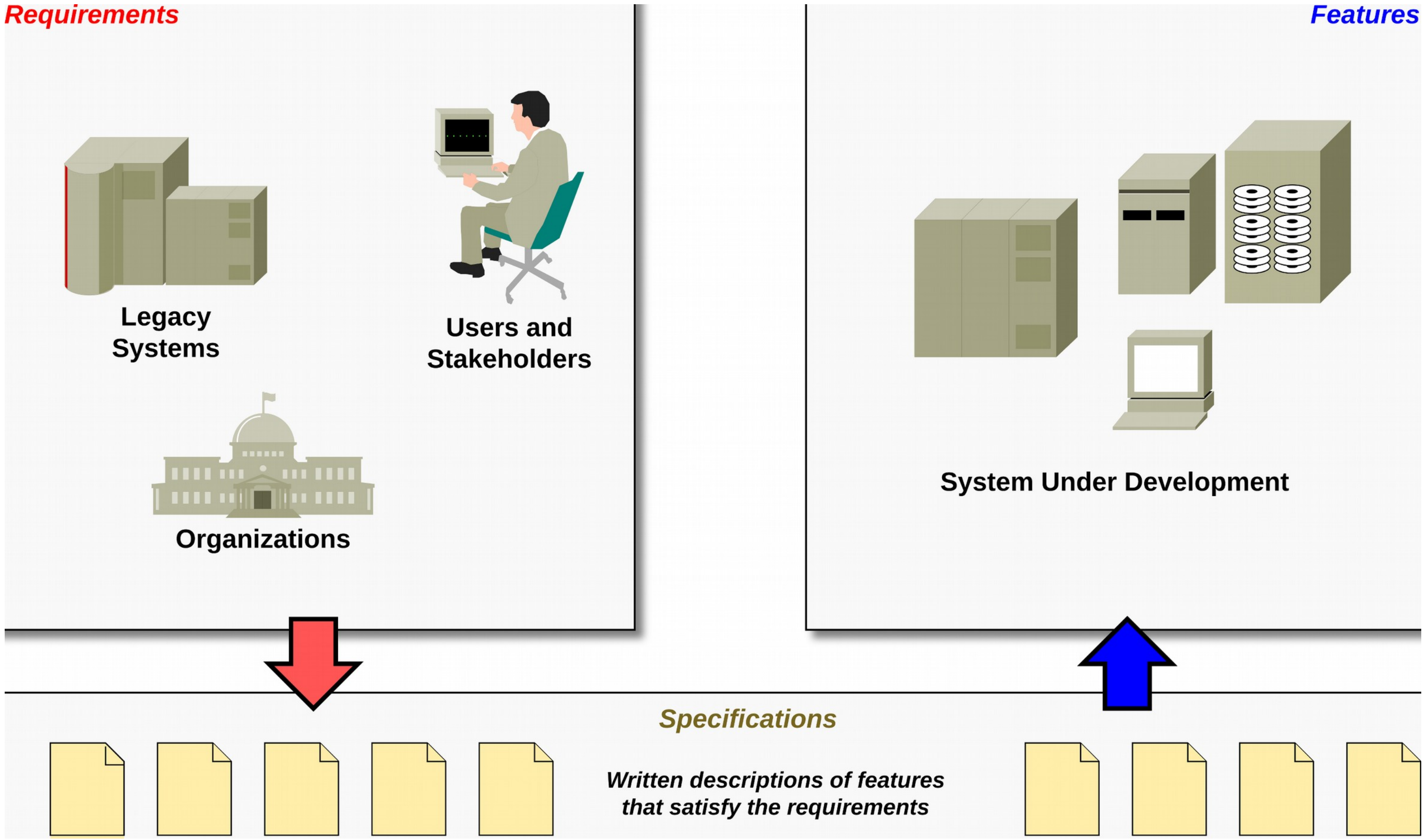
# Terminology

- A **stakeholder** is anyone:
  - who could be materially affected by the implementation of a new system or application or who has a vested interest in the project; AND
  - who has the ability, legal, political, economic or otherwise, to force the developers to protect those interests to their satisfaction.
- Stakeholders have requirements we need to deal with
- Not everyone who has requirements is a stakeholder

# Terminology

- ***Requirement***:
  - Condition or capability needed by a stakeholder to solve a problem or achieve an objective
  - "I need to be able to do this and I don’t care how it gets done"
- ***Feature***:
  - Condition or capability possessed by a system or system component to satisfy a contract, standard, specification
- ***Specification***:
  - A documented representation or description of a feature
- ***Design***:
  - A plan describing how to build features described in a specification based on available resources

# Terminology



*Requirements*

Legacy Systems

Users and Stakeholders

Organizations

*Features*

System Under Development

*Specifications*

*Written descriptions of features that satisfy the requirements*

# BA Best Practices

- BAs that “do requirements” well are characterized by common factors:
  - They have a plan and a process for getting requirements
  - The activities at each step of the process are guided by what was learned in earlier steps
  - The process is managed.  Requirements are gotten whether the stakeholders want to provide them or not
  - The stakeholders are managed to help them change their requirements so that they are realistic and lead to solutions that can be implemented within the scope of time, budget and available resources
  - The consistently follow their process
  - They continuously improve and adapt their process to new challenges
- Exactly like their counterparts in other industries

# The Standard BA Process

- Determining why we are doing the project. (Establishing project scope)
- Determining who has requirements (who the stakeholders are)
- Eliciting the requirements from stakeholders
- Analyzing the initial requirements to produce quality requirements
- Helping stakeholders modify their requirements where necessary
- Identifying potential solutions
- Producing a specification to describe the solution to be built
- Helping stakeholders modify their requirements where necessary

# BA Process Deliverables

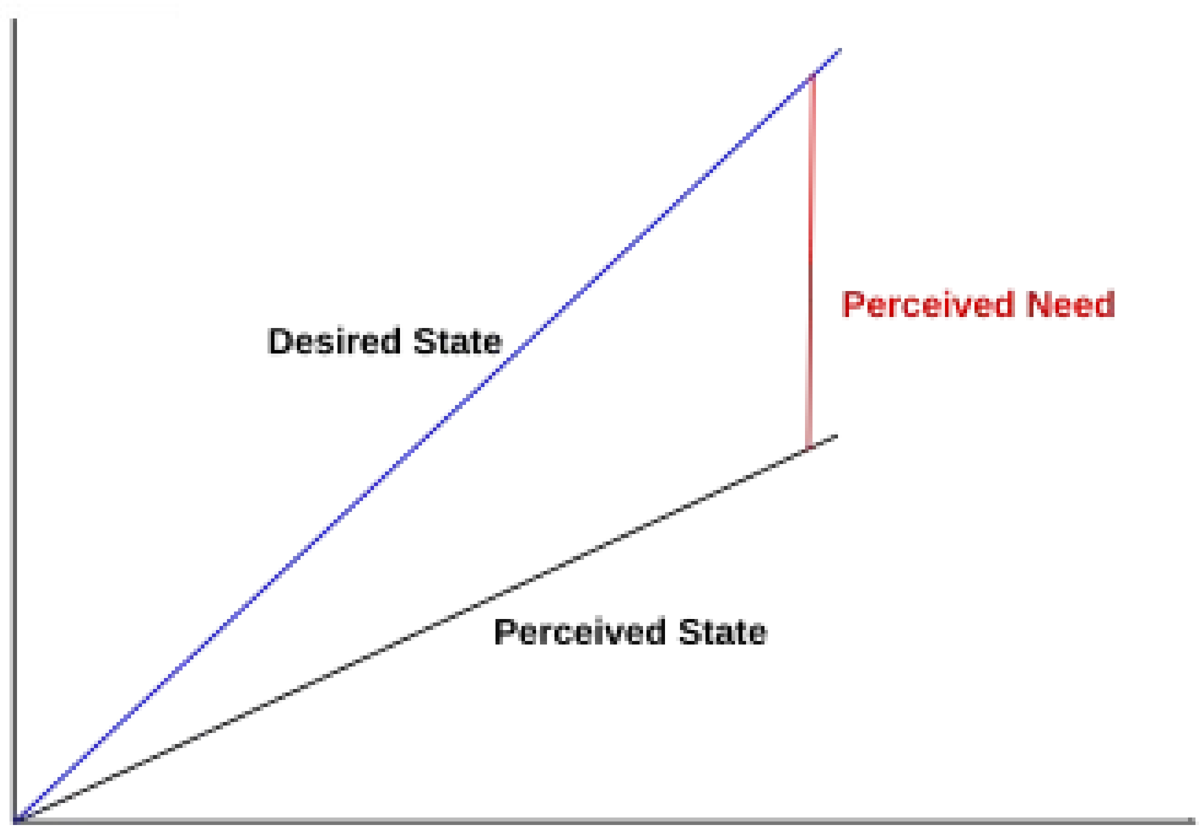
- A **system definition** that describes:
  - What the client need that the system is intended to meet is
  - Suggested solution or range of potential solutions that meet the need
  - Clearly and unambiguously identify what the scope of the project is
- A **stakeholder analysis** that describes
  - Who the stakeholders are
  - Why they are stakeholders
  - The sanctions that they can bring to bear on the project
  - A profile of each stakeholder
  - History of interaction with each stakeholder

# BA Process Deliverables

- A **unified statement of requirements** that contains:
  - A complete record of all requirement requests and their dispositions
  - A prioritized baseline of proposed features
  - A set of identified technical constraints
  - Any business processes or rules that impact a solution
- A **domain object model** that describes:
  - The entities or objects or “things” that are part of the domain
  - Serves as an input into any data models in the solution
- A **use case mode**l that describes:
  - How users will interact with the system
- A set of **non-functional requirements**

# System Definition

- Identifying the need the system should meet



# System Definition

- In terms of needs, a client is going to be in one of the following categories:
  - They perceive that a need exists, don’t know what it is.
  - They perceive a need where one does not really exist.
  - They perceive their real need accurately.
  - They perceive a need but the real need is something different than what they perceive.
- This provides the answer to two critical questions:
  - Why does the client want a new system?
  - What does the client expect the new system to do for them?

# System Definition

- Why do this?
  - It forces all the stakeholders to come to an agreement about what the project is supposed to accomplish ***before*** any work is undertaken
  - Failing to develop this document usually means the project is doomed
- The stakeholders work with the BA to develop the document
  - Differences about the project between stakeholders emerge
  - The BA is the referee/mediator/judge in this discussion

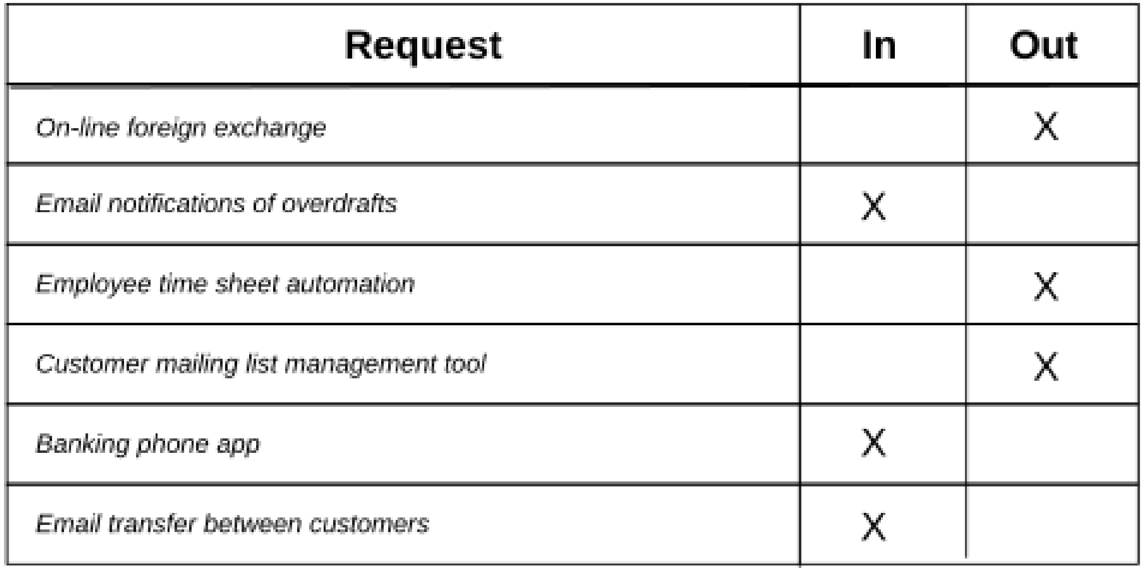
# System Definition

- Should be no more than a page
  - Think executive summary of an executive summary
  - Brevity forces stakeholders to “get to the point”
- This is not a project charter or formal document
  - This is an “excuse” to force stakeholders to agree on the project
  - It does feed into other documents like a business case
- Main parts
  - Description of the problem
  - Possible solutions (high level)
  - Key assumptions made about the project
  - Key measures for project success
    - *“The project will be a success when the we can use it to….”*

# Scope Creep

- Scope creep kills projects
  - The PMBOK® Guide describes scope creep as
  - *“adding features and functionality (project scope) without addressing the effects on time, costs, and resources, or without customer approval”*
- PMI causes of scope creep
  - Ambiguous or undefined scope definition
  - Lack of any formal scope or requirements management
  - Inconsistent process for collecting product requirements
  - Lack of sponsorship and stakeholder involvement

# In/Out List



| Request | In | Out |
|---|---|---|
| *On-line foreign exchange* |  | X |
| *Email notifications of overdrafts* | X |  |
| *Employee time sheet automation* |  | X |
| *Customer mailing list management tool* |  | X |
| *Banking phone app* | X |  |
| *Email transfer between customers* | X |  |

# How to Find Stakeholders

- Some questions to ask to help identify stakeholders
  - Who are the users of the system?
  - Who is the customer (who is paying for it)?
  - Who will maintain and support the system?
  - Whose processes will be automated or affected?
  - Whose data will be accessed or modified?
  - Who regulates the systems activities?
  - Whose standards does the system have to be in compliance with?
  - Who needs to sign off on the system and its activities?
  - Who can say NO and make it stick.
- We cannot overlook any stakeholder
- However we may use requirements from non-stakeholders for business reasons
  - Example, for good will with a community
  - For market development (providing services in other languages for example)

# Active and Passive Stakeholders

- Active stakeholders
  - Derive a direct benefit from the system or
  - Have a direct responsibility to the system
  - Normally we have to find and ask active stakeholder what their requirements are
- Passive stakeholders
  - Generally do not know about the system and do not derive any direct benefit from it
  - Become aware of the system when their requirements are not met.
  - Are concerned with regulating a class of activity like collecting sales tax or
  - Ensuring conformance to security or other standards
  - Generally passive stakeholders publish their requirements in some manner and it is up to us to find out what they are

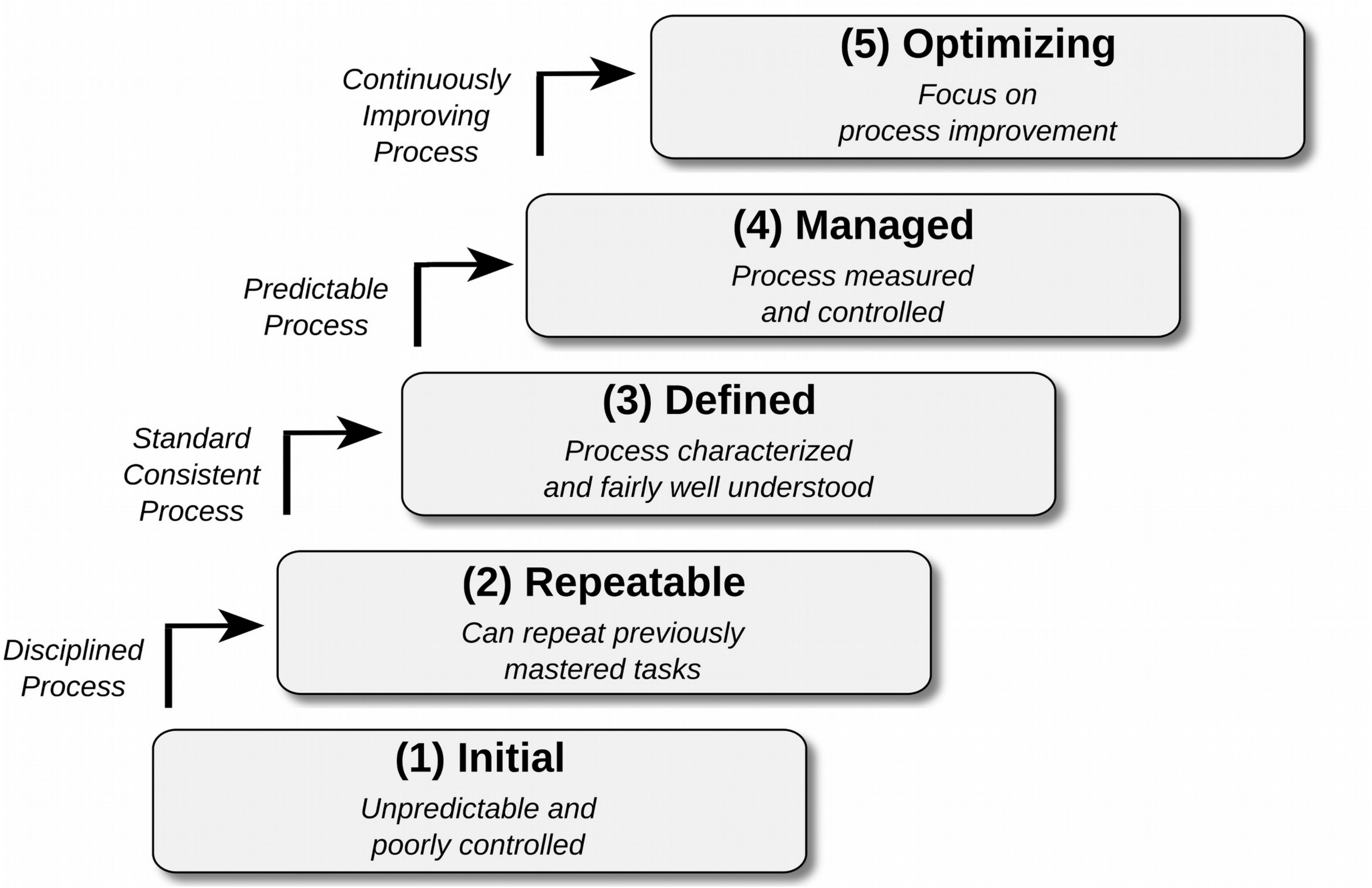
# Non-User Stakeholder Requirements

- Requirements often constrain the use of the system in some manner to preserve their interests
- These constraints often take the form of:
  - *Data capture requirements*: The system may be required to maintain audit trails or transaction records of its activities
  - *Access requirements:* The system may be required to limit or control access to certain features or functionality
  - *Functionality constraints:* The system may be required to control or manage, usually via a set of business rules, the functionality of the system.
  - *Standards Compliance:* The system may be required to perform to certain performance standards, such as encryption levels or response time requirements.
  - *Contractual and Legal Compliance:* The system may be required to implement certain features in order to be complaint with legal standards

# Process Maturity

- Software Engineering Institute
- Commissioned by US Government to find out how organizations built quality software.  They found it depended on two factors:
    - They had a defined production process.
    - They followed the process.
- Which is also true for anybody who makes anything
- Including BAs and requirements

# Process Maturity



(5) Optimizing

Focus on process improvement

Continuously Improving Process

(4) Managed

Process measured and controlled

Predictable Process

(3) Defined

Process characterized and fairly well understood

Standard Consistent Process

(2) Repeatable

Can repeat previously mastered tasks

Disciplined Process

(1) Initial

Unpredictable and poorly controlled

# Simplified CMM Level Descriptions

- ***Initial:***
  - Basically describes a start-up company – no defined processes
- ***Repeatable:***
  - Production process mastered – all projects are to spec, on time and to budget.
- ***Encultured:***
  - The process is part of the culture, it’s “the way do it here” and is  enforced by everyone in the organization.
- ***Proactive:***
  - A review based culture – everything is reviewed to ensure it is in alignment with best practices and quality metrics.
- ***Integrated:***
  - The processes have become self correcting and prevent non-optimal actions.

# Thank You for Attending

## Session 1: Mastering the Requirements Process