University of Tripoli
Faculty of Information Technology
Department of Software Engineering

# Software Requirements Analysis
# ITSE311 -- F2023

Requirements Management & Value Engineering

## Requirements Management



by: Fatima Ben Lashihar

## Introduction

- Requirements management involves identifying, documenting, and tracking system requirements from inception through delivery.
- A solid requirements management process is the key to a successful project.
- challenges to requirements management:
  - very few organizations have a well-defined requirements management process, and thus, few people in those organizations have requirements management experience.
  - the difficulty in distinguishing between user or stakeholder requirements and systems
  - organizations manage requirements differently making the dissemination (نشر) and transferability of experience and best practices difficult.
  - The difficulties in progress monitoring
  - managing changing requirements



## Introduction

- Certain requirements management approaches adapted from production engineering can be used to improve the execution of "simultaneous engineering,". What is that?!
- Most organizations do not have an explicit requirements management process in place, but this does not mean that requirements management does not occur within the organization.
- The requirements practices probably exist implicitly in the organization, but these practices are not usually documented.
- One the first steps in improving the requirements management process in any an organization is to document existing practices.

## Configuration Management and Control

- Configuration management involves the identification, tracking, and control of important artifacts of the system.
- Configuration items relevant to the requirements engineer include the individual requirements, sources of requirements, stakeholders, and the requirements specification document.
- When items are under configuration, changes to those items can only be made by those authorized to make the changes, and all of the changes are tracked, time-stamped, and version stamped.
- Configuration control involves requesting, evaluating, and approving requirements changes, approving releases.
- An important part of requirements management is disciplined configuration management control processes.



## Reconciling Differences

- One of the most important activities in requirements management is consensus building, particularly when forming or discovering:
  - Mission statements
  - Goals
  - Requirements
  - Rankings
- But achieving consensus between stakeholder groups is not easy. Due to:
  - Managing Divergent Agendas
  - Consensus Building

## Managing Divergent Agendas

- Each stakeholder has a different requirements agenda.
- For example:
  - Business owners seek ways to get their money's worth from projects.
  - Business partners want explicit requirements because they are like a contract.
  - Senior management expects more financial gain from projects than can be realized.
  - Systems and software developers like uncertainty because it gives them freedom to innovate solutions.
  - Project managers may use the requirements to protect them from false accusations of underperformance in the delivered product.



## Managing Divergent Agendas

- The smart requirements manager seeks to manage these agendas by asking the right questions up front. The following questions are appropriate:

1. What is the project request?
   a. Who wants it?
   b. Is it discretionary or nondiscretionary?
2. What is the project's purpose?
   a. If completed, what impact will the new or enhanced system have on organizational performance?
   b. On profitability?
   c. On product development?
   d. On customer retention and customer service?

## Managing Divergent Agendas

3. What are the functional requirements?
   a. What are the specific things the system should do to satisfy the purposeful requirements?
   b. How should they be ranked?
   c. What are the implementation risks?
4. What are the nonfunctional requirements, like security, usability, and interoperability?
   a. How should they be ranked?
   b. What are the implementation risks?
   c. How do you trade off functional and nonfunctional requirements?
5. Do we understand the project well enough to prototype its functionality?



## Managing Divergent Agendas

6. If the prototype is acceptable, will everyone sign off on the prioritized functionality and non-functionality to be delivered, on the initial cost and schedule estimates, on the estimates' inherent uncertainty, on the project's scope, and on the management of additional requirements?

- by asking these questions up front, hidden agendas can be uncovered and differences resolved. At the very least, important issues will be raised up front and not much later in the process.

## Consensus Building

- There are numerous approaches to consensus building including:
  - Negotiation (argument)
  - Appeals to experts
- perhaps the most celebrated consensus building technique in systems engineering is the Wideband Delphi technique.
- Wideband Delphi is usually associated with selection and prioritization of alternatives. These alternatives are usually posed in the form of a question:

for the following alternative, rate your preference according to the following scale

(5=most desired, 4=desired, 3=ambivalent, 2=not desired, 1=lease desired)



## Consensus Building

- the process is:
  1. Coordinator presents each expert with a specification and an estimation form.
  2. Coordinator calls for a group meeting in which the experts discuss estimation issues with the coordinator and each other.
  3. Experts fill out forms anonymously.
  4. Coordinator prepares and distributes a summary of the estimates.
  5. Coordinator calls for a group meeting, specifically focusing on having the experts discuss points where their estimates vary widely.
  6. Experts fill out forms, again anonymously, and steps 4 to 6 are iterated for as many rounds as appropriate.
- There will never be unanimous agreement in the Wideband Delphi process, but at least everyone involved will feel that his opinion has been considered. Wideband Delphi is a kind of win-win negotiating and can be used for other types of decision making

## Global Requirements Management

- Requirements engineering is one of the most collaboration-intensive activities in software development. Because of inadequate social contact, geographically distributed stakeholders and requirements, engineers have trouble understanding requirements issues without the appropriate tools.
- In global offshoring projects, some challenges to the requirements engineering endeavor include:
  - time delays and time zone issues,
  - the costs and stresses of physical travel to client and vendor sites when needed
  - the disadvantages of virtual conferencing and telephone.
  - Even simple email communications may lead to frequent context switching, information fragmentation, and the loss of nonverbal cues.



## Global Requirements Management

- When the offshoring takes place in a country with a different native language and substantially different culture, new problems may arise in terms of work schedules, work attitudes, communication barriers, and customer–vendor expectations of how to conduct business (recall Hofstede's metrics).
- there are vast differences in laws, legal process, and even the expectations of honesty in business transactions around the world. These issues are particularly relevant during the requirements elicitation phase.

## Global Requirements Management

- The following success factors were missing in these cases, based on an analysis of their project experiences:
  - Shared goal
  - Shared culture
  - Shared process
  - Shared responsibility
  - Trust



## Anti-patterns in Requirements Management

- In troubled organizations, the main obstacle to success is frequently accurate problem identification.
- Anti-pattern set consists of an almost even split of 28 environmental ( organizational) and 21 management anti-patterns.
  - Management anti-patterns are caused by an individual manager or management team ("the management"). These anti-patterns address issues in supervisors that lack the talent or temperament to lead a group, department, or organization.
  - Environmental anti-patterns are caused by a prevailing culture or social model. These anti-patterns are the result of misguided corporate strategy or uncontrolled socio-political forces.

## Environmental Anti-patterns

**Divergent Goals**

- The divergent goals anti-pattern exists when there are those who pull in different directions or have hidden agendas that don't align with those of the business
- If the motives really are personal, that they feel their personal success cannot come with success of the organization, radical changes are needed.
- This is most easily achieved if every stakeholder is represented in the definition and dissemination of the core mission and goals, and subsequently kept informed, updated, and represented.



## Management Anti-patterns

**Metric Abuse**

- It is the misuse of metrics either through incompetence or with deliberate malice
- When the data used in the metric are incorrect or the metric is measuring the wrong thing, the decisions made based upon them are likely the wrong ones and will do more harm than good.

## Management Anti-patterns

- Suppose a fire control system for a factory is required to dispense fire retardant in the event of a fire. Fire can be detected in a number of ways—based on temperature, the presence of smoke, the absence of oxygen on the presence of gases from combustion and so on. So, which of these should be measured to determine if there is a fire? Selecting the wrong metric can lead to a case of metrics abuse.

- The solution or refactoring for metrics abuse is to stop the offending measurements. Measuring nothing is better than measuring the wrong thing. When data are available, people use them in decision making, regardless of their accuracy.



## Value Engineering

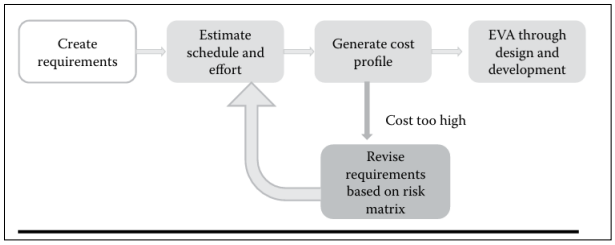
## What Is Value Engineering?

- There is a fundamental relationship between time, cost, and functionality. Project managers sometimes refer to this triad as the three legs of the project managements tool.
- When dealing with the affordability during requirements elicitation, analysis, agreement, and negotiation, both the vendor and customer have to balance certain factors:
  - determine the functional and nonfunctional features or performance of the system
  - A schedule has to be determined and agreed upon.
  - identify various risk factors in both the actual system features and in the processes to develop the system
- As a result, the requirements, schedule, and risk factors determine the cost to produce, and thus, the price to the customer. The vendor is entitled to a fair profit, but at the same time, the customers' expectations need to be managed.



## What Is Value Engineering?

- Making such estimations is not easy to do accurately at the early stages of a project, such as during the requirements elicitation activities.
- The activities related to managing expectations and estimating and managing costs are called value or affordability engineering
- Value engineering occurs throughout the system life cycle and is typically considered a project management activity.
- For the requirements engineer, value engineering has to take place to help manage customer expectations concerning the final costs of the delivered system and the feasibility or infeasibility of delivering certain features.
- The best time to conduct the cost analysis is at the time when the systems-level requirements are being put together. It is at this time that better cost estimates are available, and this is a time when trade-off decisions can be discussed more successfully with the customer, using some of the negotiating principles previously mentioned.




Simple affordability engineering for requirements

1. working with the customer and other stakeholders we elicit, analyze, and create requirements using standard approaches.
2. using some estimating techniques, such as some constructive cost modeling tool to calculate the effort needed to produce those requirements.
3. generating a cost profile for these requirements based on the effort estimates just calculated. If the cost of these requirements is too high, then we need to revise the requirements set.
4. recalculating the effort and cost for the revised requirements set until the vendor and customer are satisfied.
5. using standard project monitoring techniques to ensure that as requirements are built out, they meet their target cost structure.



## Challenges to Simple Cost vs. Risk Analysis

- Computing true costs of requirements can be quite difficult early on
- Take into account any dependencies between requirements may be a challenge
- Different customers and stakeholders will rate requirements risk differently. But they will also decompose functionality differently. For example, one customer might view a requirement for a certain style user screen to be a minor task, while another might view the implementation of this feature as a major undertaking.
- Customers and other stakeholders have different levels of commitment to a project, and hence will invest more or less thought into their inputs.
- Those participating in the requirements elicitation process will have different personal involvement. Hence, influence the quality of their participation in requirements elicitation and agreement.

## Challenges to Simple Cost vs. Risk Analysis

- The effects of differing personal agendas of the participants. For example, a customer representative who is leaving the company soon will care less about the correctness of a feature description than the prospective manager of the project.
- Cost-based requirements engineering can encourage individuals to give an excuse if the project does not go as planned. You can identify these individuals, usually, because they are unwilling to attach their name to certain documentation, or they use hedging words during requirements negotiation.



## Estimating Using COCOMO and Its Derivatives

- COCOMO (constructive cost model): this software is commercially available and can even be found on the Web for free use. It is based on an estimate of lines of code - which are not easily estimated at the time of requirements engineering-, modified by a number of factors.
- WEBMO is a derivative of COCOMO that is intended specifically for project estimation of Web-based projects, where COCOMO is not always as good. WEBMO uses the same effort and duration equations as COCOMO, but is based on a different set of predictors. the net result of a WEBMO calculation is a statement of effort and duration to complete the project in person-months and calendar months, respectively.
- COSYSMO (COnstructive SYstem engineering MOdel): ) is a COCOMO enhancement for systems engineering project cost and schedule estimation. It is intended to be used for cost and effort estimation of mixed hardware/ software systems based on a set of size drivers, cost drivers, and team characteristics.

## Estimating Using Function Points

- Function points were introduced as an alternative to product metrics based on source line count.
- The basis of function points is that as more powerful programing languages were developed, the number of source lines necessary to perform a given function decreased. Paradoxically, however, the cost/LOC measure indicated a reduction in productivity, as the fixed costs of software production were largely unchanged
- The solution to this effort estimation paradox is to measure the functionality of software via the projected number of interfaces between modules and subsystems in programs or systems. A big advantage of the function point metric is that it can be calculated during the requirements engineering activities.



## Estimating Using Function Points

- Function Point Cost Drivers: it takes into account the following five software characteristics for each module, subsystem, or system represent the function points or cost drivers: Number of inputs to the application, Number of outputs, Number of user inquiries (Q), Number of files used, Number of external interfaces (X)
- Feature Points: they are an extension of function points. address the fact that the classical function point metric was developed for management information systems and therefore is not particularly applicable to many other systems, such as real-time, embedded, communications, and process control software. The motivation is that these systems exhibit high levels of algorithmic complexity, but sparse inputs and outputs. the feature point metric is not widely used.
- Use Case Points: it allow the estimation of an application's size and effort from its use cases. They are beneficial for effort estimation in those projects where the requirements are created predominantly via use cases.

The End……