



<p><i>Dr. Aldo Dagnino</i> <i>ABB, Inc.</i> <i>US Corporate Research</i> <i>Center</i> <i>October 21<sup>st</sup>, 2003</i></p>	<p><b>Requirements Engineering</b></p>
	

## **Class Objectives**

- Students will be able to define the two process areas associated with the Requirements Engineering process
- Students will be able to describe the difference among functional requirements, non-functional requirements, fit criteria, and constraints
- Students will be able to document software requirements

## Requirements Engineering

- Requirements Development Process
  - The purpose of the requirements development process is to produce and analyze customer, product, and product-component requirements
- Requirements Management Process
  - The purpose of the requirements management process is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products

Chrissis, et al. (2003) "CMMI: Guidelines for Process Integration and Product Improvement", Addison-Wesley, ISBN 0 321 15496 7

## Requirements Development Overview (1)

- This process area describes three types of requirements: customer requirements, product requirements, and product-component requirements. Requirements are the basis for architecture and design. The development of requirements includes the following activities:
  - Elicitation, analysis, validation, and communication of customer needs, expectations, and constraints to obtain customer requirements that constitute an understanding of what will satisfy stakeholders
  - Collection and coordination of stakeholder needs
  - Development of lifecycle requirements of the product
  - Establishment of customer requirements
  - Establishment of initial product and product-component requirements consistent with customer requirements

Chrissis, et al. (2003) "CMMI: Guidelines for Process Integration and Product Improvement", Addison-Wesley, ISBN 0 321 15496 7

## . . . Requirements Development Overview (2)

- The Requirements Development process area includes three **Specific Goals** (SGs) according to the Capability Maturity Model Integration (CMMI):
  1. **Develop Customer Requirements**
    - Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements
  2. **Develop Product Requirements**
    - Customer requirements are refined and elaborated to develop product and product-component requirements
  3. **Analyze and Validate Requirements**
    - The requirements are analyzed and validated, and a definition of required functionality is developed

Chrissis, et al. (2003) "CMMI: Guidelines for Process Integration and Product Improvement", Addison-Wesley, ISBN 0 321 15496 7

## Requirements Management Overview (1)

- The purpose of this process area is to manage all requirements received or generated by the project, including both technical and non-technical requirements.
  - Agreed-on set of requirements must be managed to support the planning and execution needs of the project.
  - When a project receives requirements from an approved requirements provider, these requirements are reviewed to resolve issues and prevent misunderstandings before they are incorporated into the project plan.
  - Commitment to the agreed requirements is received from project participants.
  - Changes to the requirements must be managed as they evolve and any inconsistencies must be identified.
  - Management of requirements involves as well to document requirements changes and rationale, and to maintain bi-directional traceability between source requirements and all product and product-component requirements.

Chrissis, et al. (2003) "CMMI: Guidelines for Process Integration and Product Improvement", Addison-Wesley, ISBN 0 321 15496 7

## . . . Requirements Management Overview (2)

- The Requirements Management process area includes one **Specific Goal** (SG) according to the CMMI:

### 1. **Manage Requirements**

- Requirements are managed and inconsistencies with project plans and work products are identified. Current and approved set of requirements are maintained over the lifecycle of the project by:
  - Managing all changes to the requirements
  - Maintaining the relationships among the requirements, the project plans, and the work products
  - Identifying inconsistencies among the requirements, the project plans, and the work products
  - Taking corrective action

Chrissis, et al. (2003) "CMMI: Guidelines for Process Integration and Product Improvement", Addison-Wesley, ISBN 0 321 15496 7

## Life Cycle Methodology

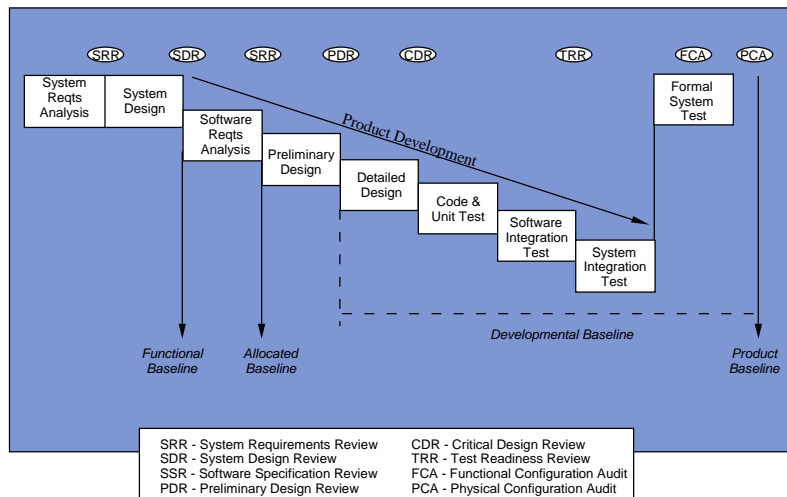
- A Life Cycle Methodology deals with the order in which the activities, methods, practices, and tools are applied to the development and maintenance of software
- Identifies the major activities which occur in the development and maintenance of a software system
- Orders the activities into sequenced stages
- Identifies the results of the stages and the criteria for progressing from one stage to the next
- Is used for planning, scheduling, monitoring, and controlling a project

## Basic Life Cycle Phase Tasks

Phase	Tasks
Proposal	understand the customer's needs analyze requirements, develop response develop proposal & cost packages
Requirements	define functional / performance / design requirements design system architecture with formal division for hardware, software, and procedure analyze system requirements allocated to software and create specification
Design	define architecture of, and communication among, the software components (functions & interfaces) define algorithms and data structures for lower-level components
Implementation	code and unit test
Test	test against software high-level design (software component interactions and interfaces) test against requirements allocated to software test system requirements (subsystems interfaces and external interfaces)
Maintenance	update system (includes all above tasks)

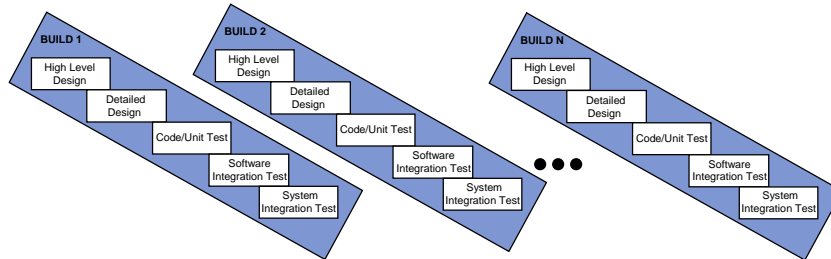
**Maintain consistency backwards and forwards across work products**

## Sequential (Waterfall)



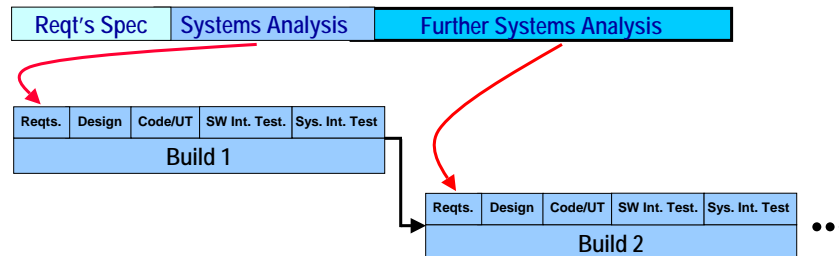
## Incremental

- Delivers some of the features of the final system in a preliminary release - a usable core system
- Delivers additional features as upgraded releases which include the previous features.
- All requirements set up front and allocated to different releases



## Evolutionary

- Allows new requirements to be incorporated
- Provides control points for injecting new technology
- Provides opportunities for customer review and confirmation of marketing/customer expectations
- Does not require all requirements to be set in the beginning

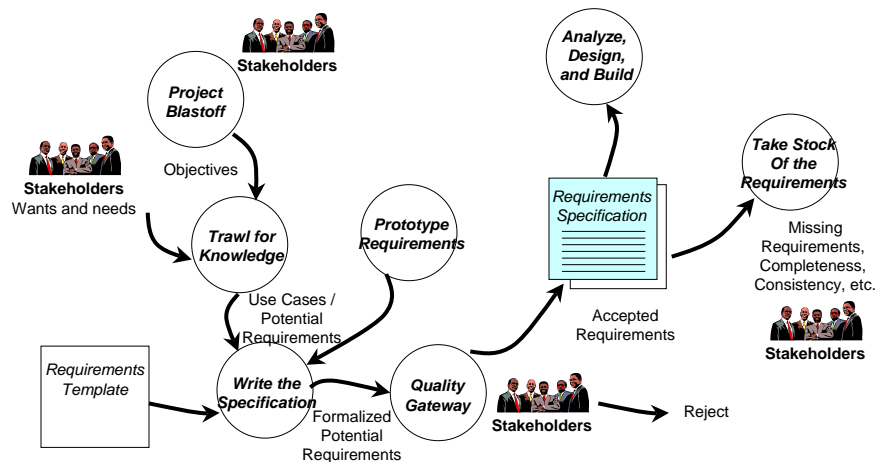


## What are Requirements?

- Requirements are the “elements” that the requirements analyst should discover before starting to build a system. A requirement represents “something” that the system must do or a quality that the system must possess.
  - Functional requirements
  - Non-functional requirements
  - Constraints

## Volere Requirements Process Model

- Generic requirements gathering and specification process to explore, capture and communicate the requirements. The Volere process provides a guide for how to discover, document and write testable requirements.



## System Purpose Statement

- Describes the reason behind building the system
- The system purpose statement represents the highest-level customer requirement
- All other requirements gathered must contribute to achieve the system purpose
- All requirements will be tested against the statement on purpose
- Consensus on the system purpose statement needs to be reached during the project blast-off stage
- The system purpose statement must solve a problem and provide a business advantage
- Sometimes the system has more than one purpose statement

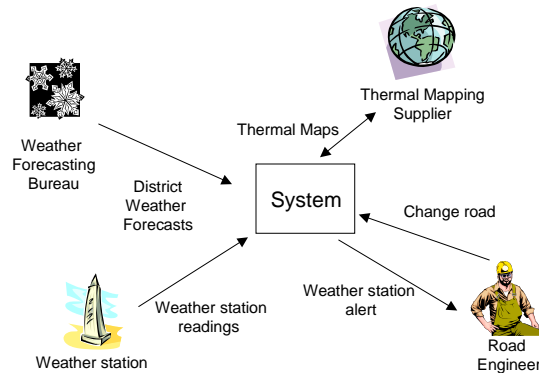
*Purpose: to accurately forecast road freezing times and dispatch de-icing trucks*

## Aspects of the System Purpose Statement

- Purpose – description of what the system is to do
- Advantage – what business advantage does the system provide?
- Measurement – how is the advantage measured?
- Reasonableness – is the product construction effort greater than the advantage?
- Feasibility – can the system achieve the expected measure?
- Achievability – does the development organization have the skills to build the product and operate it?

## System Context

- The system context diagram shows the boundary of the system and its adjacent systems
- Named arrows represent data flows and directions of flows
- The adjacent systems represent the domains with which the system needs to interact



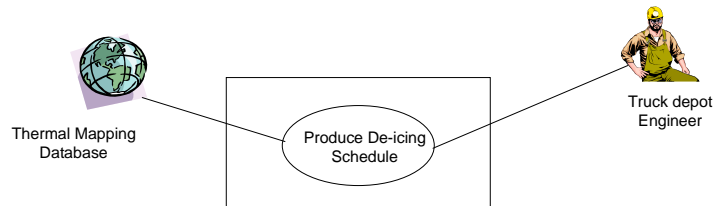
Robertson, S. and Robertson, J. (1999) "Mastering the Requirements Process", Addison-Wesley. ISBN 0 201 36046 2

## Trawling for Requirements

- The requirements analyst translates the user's/customer's needs into a system specification
- The requirements analyst must understand the current user's work, and determine the work that the user and customer requires to do in the future
- Requirements analyst instigates requirements trawling
- Users and system relevant stakeholders collaborate with requirements analyst to gather the requirements
- Some techniques for requirements trawling
  - Apprenticing – learn job by observation and model system
  - Structures and patterns – abstract structure and pattern of work
  - Interview users – used as a complement
  - Workshops – brainstorm sessions with relevant stakeholders – mind mapping
  - Video
  - Electronic requirements gathering – via e-mail, internet, surveys
  - Document reviews
  - Cards, spreadsheets, or other light-weight approaches

## Event-driven Use Cases

- Work performed by the system in response to a business event. The *use case* is a convenient way of identifying a user and a group of requirements that carry out a specific task for that user.



- Task that the actor describes in his/her own language at too high level to describe details about system's capabilities

## Produce Road De-icing Schedule Use Case: Steps

- Suggested desired outcome for this use case:
  - System accepts scheduling date and district identifier from engineer
  - System fetches the relevant thermal maps
  - System uses thermal maps, district temperature readings and weather forecasts to predict temperatures for each part of the district
  - System determines which roads are likely to freeze and when they are likely to freeze
  - System schedules available trucks from the depots responsible for the freezing roads
  - System advises the engineer of the schedule

## Functional Requirements Derived

- System accepts scheduling date and district identifier from engineer
  - *The system shall accept the scheduling date*
  - *The system shall warn if scheduling date is neither today nor within the next two days*
  - *The product shall accept a valid district identifier*
  - *The product shall confirm that the district selected is the one wanted by the engineer*
- Notice
  - the level of detail: they can be verified, they are enough to describe the use case
  - Reduce ambiguity to ensure “correct” meaning
  - Requirements need the so called “fit criteria”

## Functional Requirements

- Functional requirements represent the capabilities that the product must have to achieve its purpose – an action that the system must take if it is to provide useful functionality for its user.

*The system shall record air temperature readings and humidity readings*

*The system shall accept a scheduling date*

*The system shall accept a valid district identifier*

*The system shall confirm that the district selected is the one wanted by the user*

## **Non-Functional requirements**

- Describe the experience that the user has while he/she does the work
- They describe the characteristics of the work that are represented by the use case or the functional requirements

## **Non-functional Requirement Types**

- Look and feel requirements
- Usability requirements
- Performance requirements
- Operational requirements (operating environment)
- Maintainability and portability requirements
- Security requirements
- Cultural and political requirements
- Legal requirements

## Non-Functional Requirements

- Non-functional requirements represent the product qualities that the system must possess (i.e. look and feel, usability, performance, security, maintainability, cultural and political, legal, etc.).

*The system shall calculate change in road topography in 1.5 seconds*

*The system shall provide a graphic description and colorful view of all roads in a district*

*The system shall comply with the Windows NT guidelines*

*The system shall be easy to use*

*The system shall comply with ISO 9000 Certification*

## Fit Criteria

- “Fit” means that a solution completely satisfies the defined requirement
- Need to attach a quantification to the requirement
- The quantification of the requirement is its fit criterion
- The *fit criterion* may quantify the behavior, the performance, or some other quality of the requirement
- *Fit criteria* apply to both functional and non-functional requirements
- Analyze requirement description and determine requirement rationale to find the appropriate scale of measurement for *fit criteria*

## Requirements with Fit Criteria Examples

- Functional Requirement
  - *Description:* The system shall record the weather station readings
  - *Rationale:* so readings are not lost
  - *Fit criterion:* The recorded weather station readings shall match the readings sent by the weather station
- Non-Functional Requirement
  - *Description:* The system shall be user friendly
  - *Rationale:* so new users can learn system fast
  - *Fit criterion:* new users shall be able to add, change and delete roads within 30 minutes of their first attempt at using the product

Robertson, S. and Robertson, J. (1999) "Mastering the Requirements Process", Addison-Wesley. ISBN 0 201 36046 2

## Constraints

- Constraints are typically viewed as global requirements. They apply to the entire system and preferably defined before beginning the work on gathering the requirements. Constraints represent global issues that shape the requirements.

*The system must run in a hand-held device*

*The system will be deployed in a noisy environment*

*The system must be dust resistance*

*The user will be standing up while operating the system*

## Exercise

- Write one functional requirement, one non-functional requirement and their fit criteria, as well as one constraint associated with their project (please do this exercise independently)
- 10 minutes