

Projektowanie oprogramowania systemów

System software design

Politechnika Gdańska

Katedra systemów multimedialnych

- dr inż. Arkadiusz Harasimiuk
- email: arekh@multimed.org

PLAN OF THE COURSE AND ASSESSMENT

- Lecture:
 - first half of semester
 - finished by exam
- Project
 - second half of semester
- Assessment:
 - exam – 50%
 - project – 50%

Agenda

- System Development Life Cycle (SDLC)
- Business Analysis
- Analysis – methods
- Solutions Architecture
 - What is IT architecture
 - Architecture development method
 - Architecture governance
 - Reference models
 - Content metamodel
- Specific dedicated solutions
 - Mobile
 - Cloud
- DevOPS
- Developers tools
- Project management
- Design patterns, tests

SDLC

SDLC

the acronym for

Software
Development
Life Cycle.

SDLC

Literature

[https://www.tutorialspoint.com/sd
lc/index.htm](https://www.tutorialspoint.com/sd
lc/index.htm)

[https://www.w3schools.in/sdlc-
tutorial/software-development-
life-cycle-sdlc/](https://www.w3schools.in/sdlc-
tutorial/software-development-
life-cycle-sdlc/)

SDLC

- Software Development Life Cycle - a process followed by software industries to design, implement and test the software products.

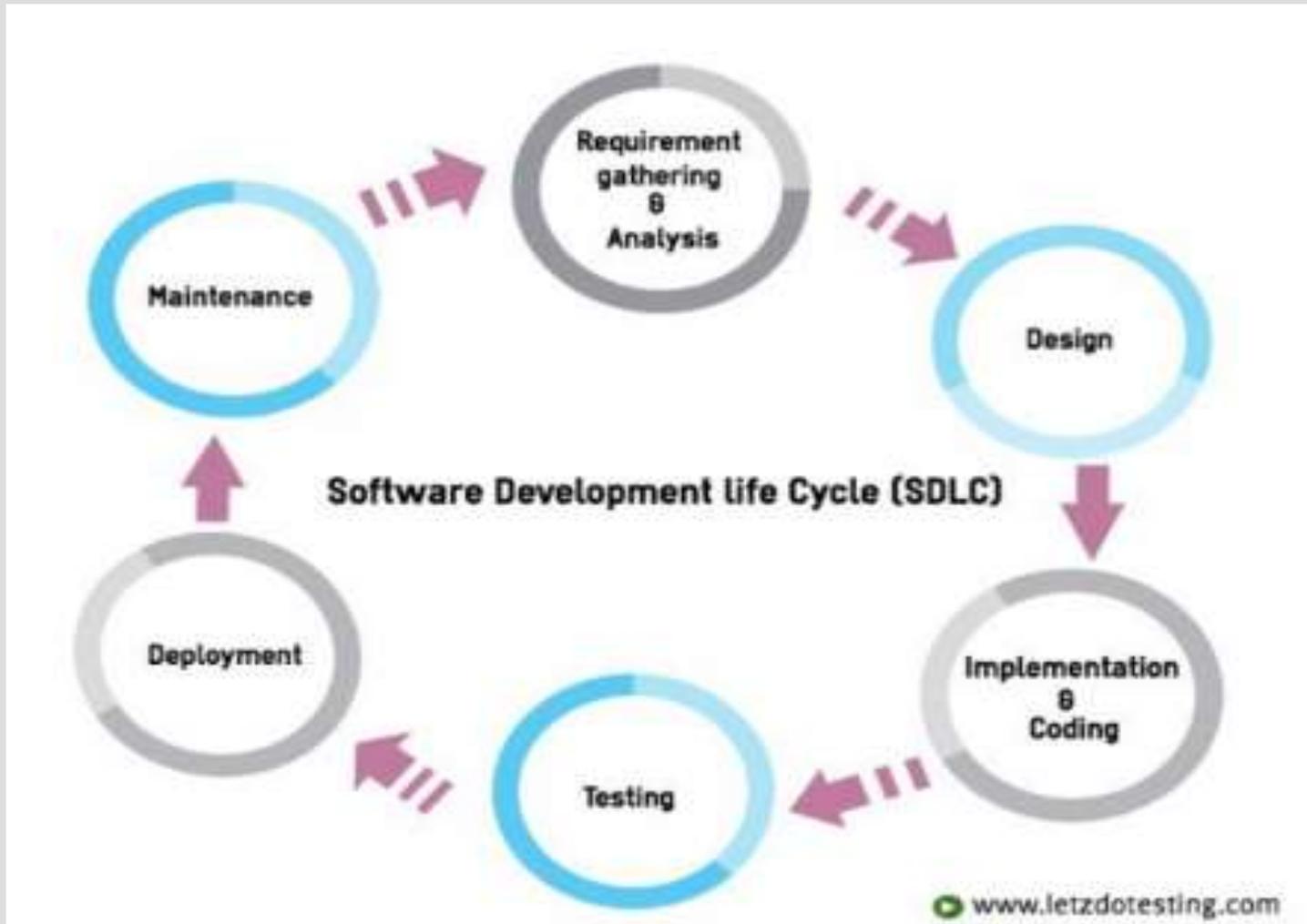
SDLC process - to ensure **quality products** are delivered to the customers and consumers within the **planned budget** and **time estimates**.

An entire team of Software Engineering will be highly involved in the development of the high standard software. Whole Life Cycle defines and improves the overall software quality and development process. SDLC has six stages while each phase has its own deliverables which serve as the input for the next phase.

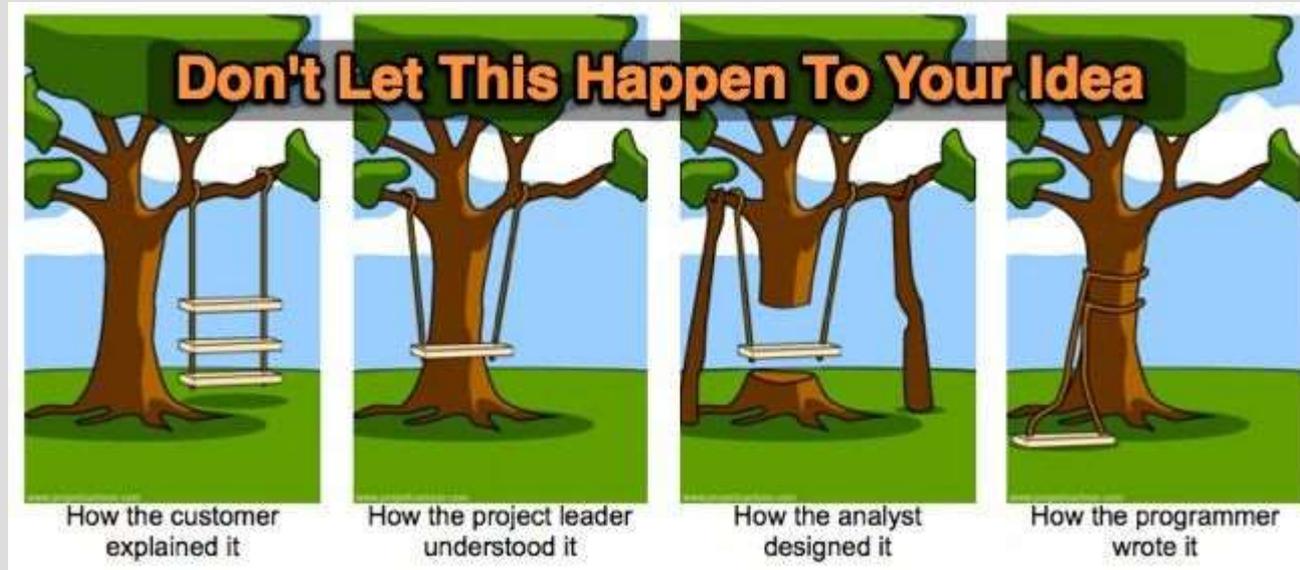
Stages of SDLC

1. Requirement gathering & Analysis
2. Design
3. Implementation & Coding
4. Testing
5. Deployment
6. Maintenance

SDLC



SDLC



SDLC

- **Requirement gathering & Analysis**

- Requirement gathering - fundamental phase of software development. Business requirements and specifications are gathered.

Top level management of the company like project managers, directors, sales, consulting, marketing and other stakeholders are mainly involved in drafting the requirements of the software.

Analysis should give Answers to the general questions:

- What is the purpose of the system?
- How will the system be used?
- Who will use the system?
- What is the required data?
- What is the outcome/usage of the system?

Once the requirements are gathered, it will be analyzed by a group of individuals to make sure the possibility of incorporating software development with the requirements gathered.

Deliverable:

Business Requirement Documentation (BRD), Software Requirement Specifications (SRS), Technical Requirement Specifications documents are created which serves as the input for next phase- Design

SDLC

Design

The design phase is the second phase of software development.

System Architecture is designed during this phase.

Design architects, Business Analysts - responsible for drafting the design of the software. High-level design (HLD) and Low-level design (LLD) are designed based on the requirements gathered in analysis phase.

Example of design documents that helps in developing the software effectively.

- Use case Diagram
- Class Diagram
- Entity-Relationship (ER) Diagram
- Component Diagram
- Workflow diagram
- Activity Diagram

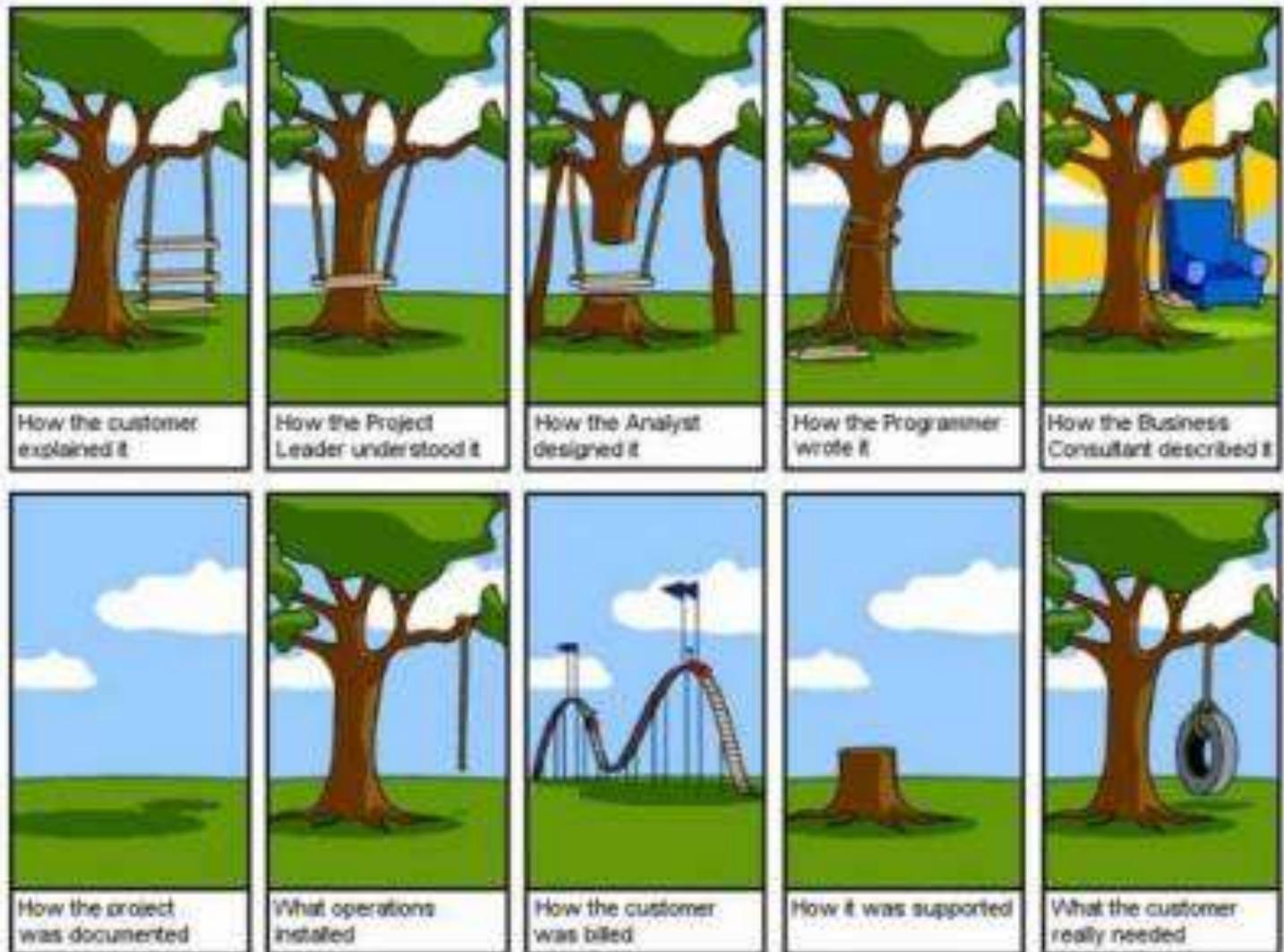
Design documents - input for the implementation/coding phase.

Testers - test strategy during this phase like what to test and how to test

Deliverable:

System Architecture, HLD, LLD, Detailed Design Specifications (DDS) - created which serves as the input for next phase – Coding.

SDLC



SDLC

Implementation/Coding

Coding is third phase of software development.

Detailed design specifications - input by developers to build the software product. The main focus of this phase - development.

Entire design will be broken into **modules** and developers will work on **individual modules**, then they will **integrate the separate modules** into one system finally.

Coding standards and guidelines are followed while developing the product. It differs between the companies.

Coding languages like Python, JAVA, PHP, C++, C#, ASP.NET - used to develop the software and languages are chosen based on the product to be developed.

When two developers working on the same software at the same time, code merge is accomplished without much hassle by following proper coding guidelines.

Deliverable:

Working software is developed which serves as the input for the next phase –Testing.

SDLC

Testing

Testing is the fourth phase of software development. Working software is used for testing. The main focus of this phase is to ensure quality and to validate the tests against the customer requirements to make sure the product satisfies the customer needs. Test cases and test scripts are used for testing to validate the requirements. Following types of testing are done but the list is not limited:

- System testing
- System Integration testing
- Component testing
- Component Integration testing
- Performance testing
- Usability testing
- Acceptance testing

Software product bugs, issues, defects are tracked, reported and retested until the product achieves quality.

Deliverable:

Test Summary Report, Test results, QA plan, Revised bugs list, User Acceptance test are submitted which serves as the input for the next phase – Deployment

SDLC

Product development from an IT failures perspective



How the customer explained it



How the project leader understood it



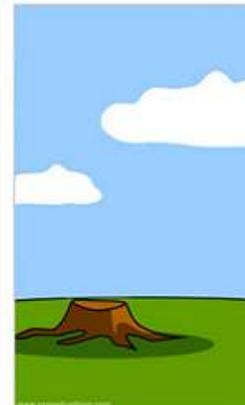
How the analyst designed it



What operations installed



How it was documented



How it was supported



How they advertised the open source version



How they applied open source patches



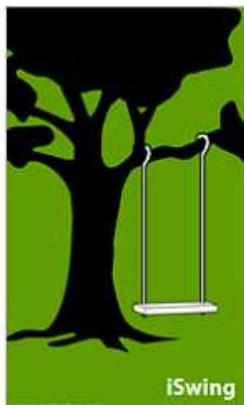
How the business consultant described it



How the programmer wrote it



What the beta testers received



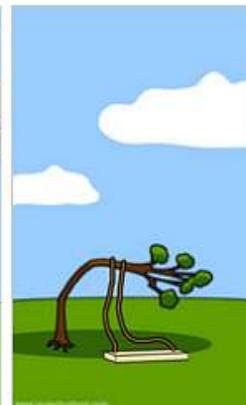
What marketing advertised



What the customer really needed



How it performed under load



The disaster recover plan

SDLC

Deployment

Deployment is the fifth phase of software development. Once the product is successfully tested, the software product can be released or deployed to the customers for their usage. If any changes or enhancements required for the software is also conveyed to the team so that the changes will be incorporated in their subsequent releases.

Beta testing is done at the customer's site before the final deployment to ensure the product meets customer requirements.

Beta testing feedbacks and issues are reported back to the engineering team to ensure they address them sooner or later.

Once the issues are fixed, then the product is finalized and will be ready for final deployment.

Deliverable:

Deployed software, Customer's review, Live Production environment which serves as the input for the next phase – Maintenance.

SDLC

Maintenance

Maintenance is the sixth phase of software development. Once the product is used by customers, the issues faced by them are addressed during the maintenance phase. Support provided for the deployed software is called Maintenance. Handling of bugs those are missed in the testing phase are taken care of during this phase.

If the deployed system is modified for the new needs, then the existing system should be maintained. If existing system cannot be maintained once the new needs are being added, then the whole software system should be re-designed to accommodate the new changes. The performance of the system is also monitored during this phase. Maintenance phase ensures whether the customer is satisfied with the product that has been delivered.

Deliverable:

Updated version of the product, Code maintenance, Live system

SDLC

System Development Life Cycle (SDLC)

- - Complex process of developing and maintaining an information system
 - Key Phases
 - o **Planning**
 - Figure out why an information system needs to be created and how it would be built
 - o **Analysis**
 - Figure the four w's
 - Who will be the users of the system?
 - What will the system do?
 - Where will the system be utilized?
 - When will it be used?
 - Study and structure requirements
 - o **Design**
 - How system will operate in terms of hardware and software.
 - Logical Design*
 - Functional features of system are described independently of software usually in a model.
 - Ex: a diagram on paper or digital diagram
 - Physical Design*
 - Transformation of logical design into tech-specific details which is used and construct the system
 - o **Implementation**
 - System is built, bought and installed
 - Longest and most expensive phase
 - Information system is:
 - Coded (specifications are programmed into the system)
 - Tested (testing for bugs and fixing them, critical step)
 - Installed (switching between the old and new system)
 - Supported (create a support plan)

platform

to program

SDLC

Zachman's framework

Perspective		Aspect		
Planner	Establishes scope and purpose	WHAT	Things	Databases
Owner	Recipient/user of end product or service	HOW	Processes	Software,
Designer	In-between who balances owner's desires with what is achievable	WHERE	Connectivity	Hardware, networks
Builder	General contractor	WHO	People	Organizations,

The Zachman Framework is an enterprise ontology and is a fundamental structure for Enterprise Architecture which provides a formal and structured way of viewing and defining an enterprise. The ontology is a two dimensional classification schema that reflects the intersection between two historical classifications.

SDLC

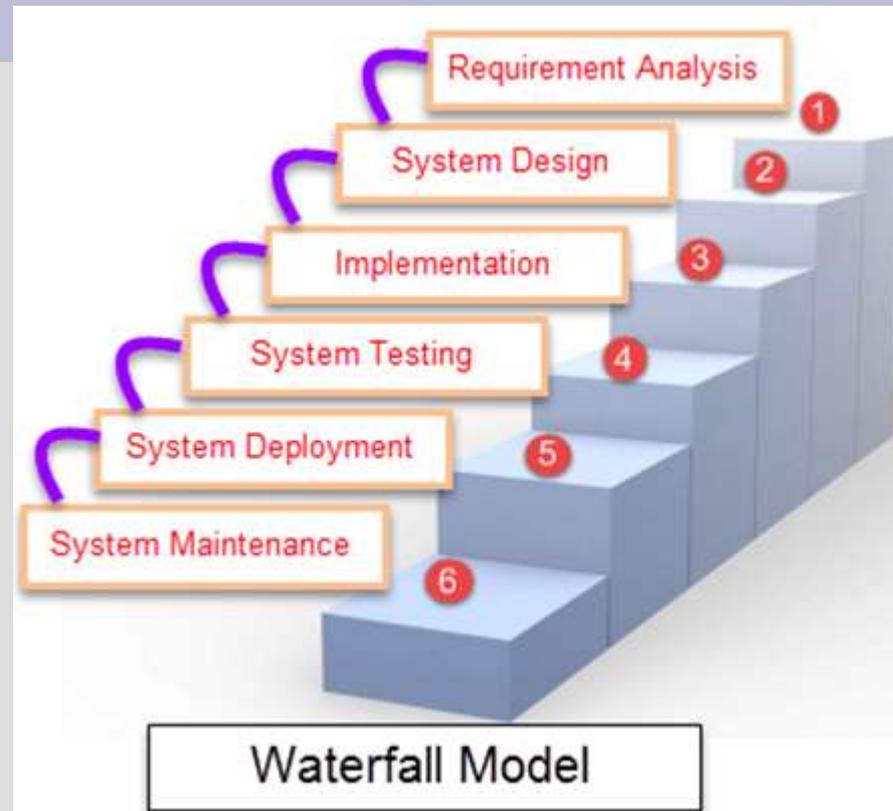
Types of SDLC Models

SDLC Models are developed based on the phases of SDLC. The difference between each model would be the sequence in which they happen and the interaction between phases.

Common and most popular SDLC Models are

- Waterfall Model
- Agile Model
- Iterative Model
- V-Model
- Spiral Model
- Big Bang Model

SDLC Waterfall Model

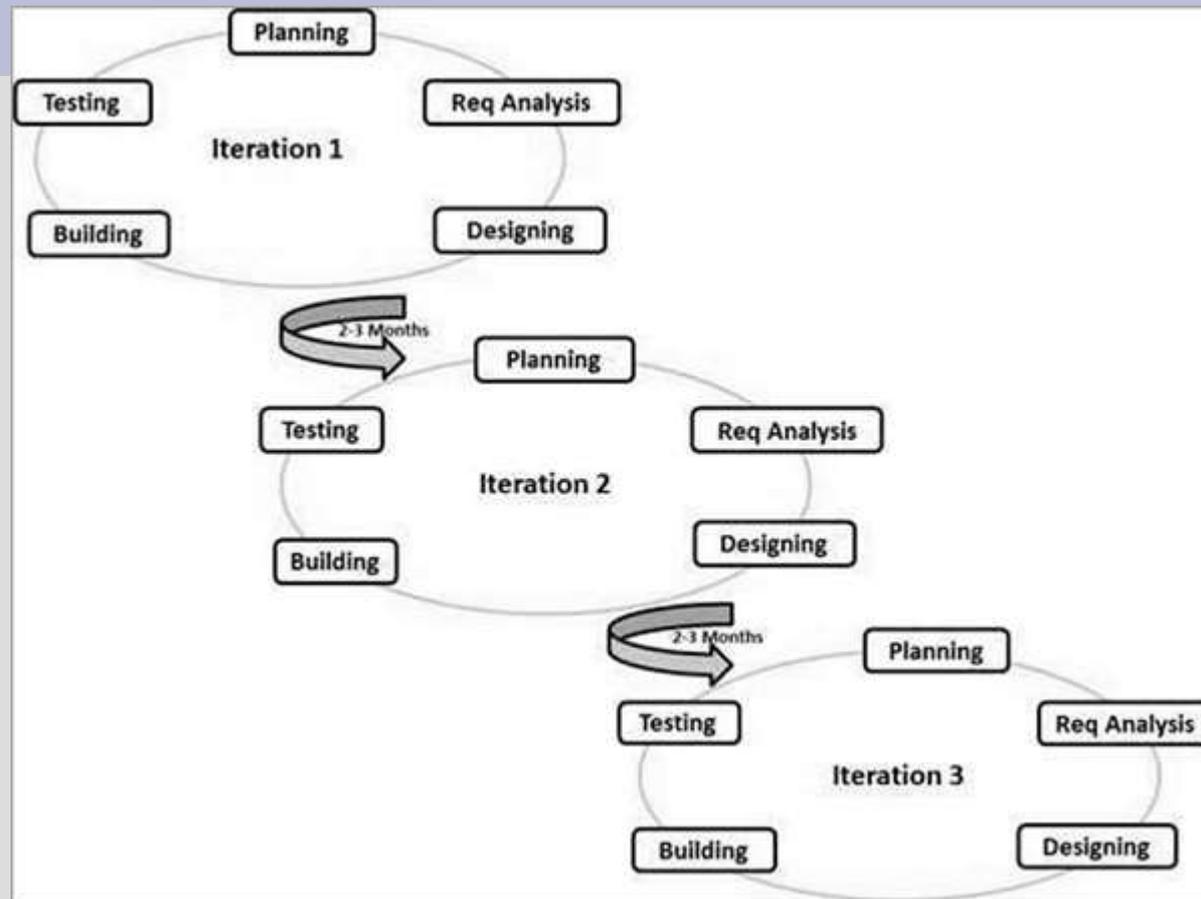


The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

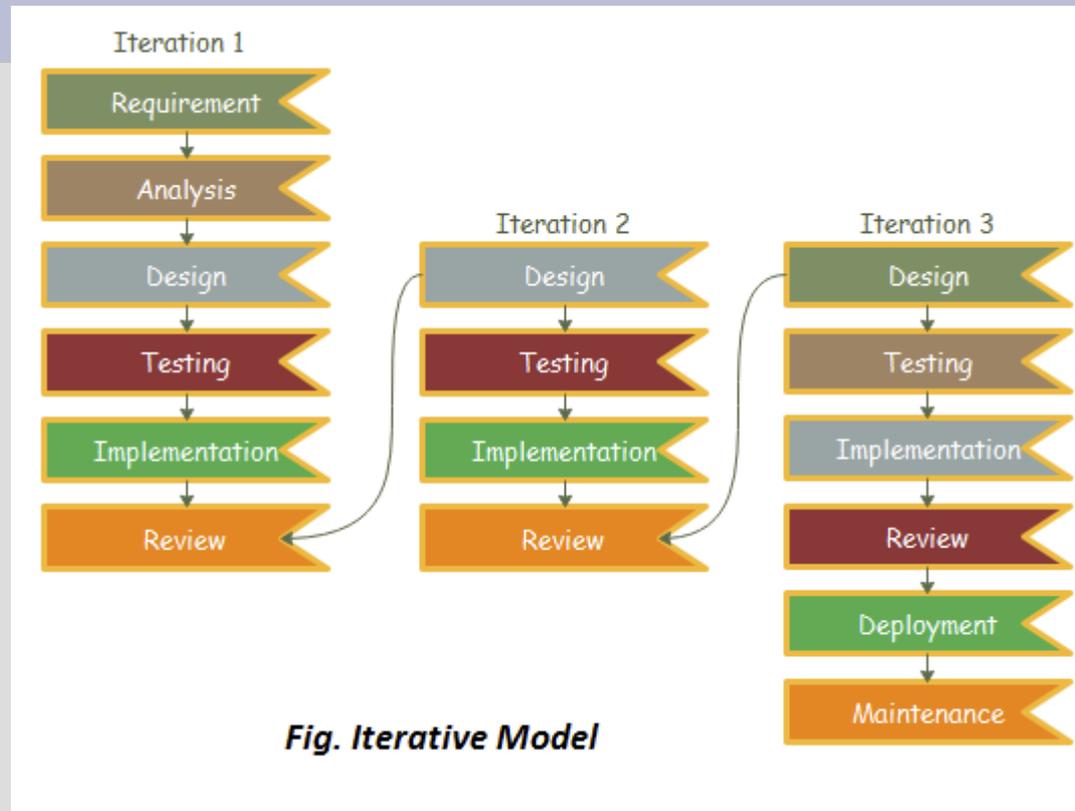
SDLC Agile Model



Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas |

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

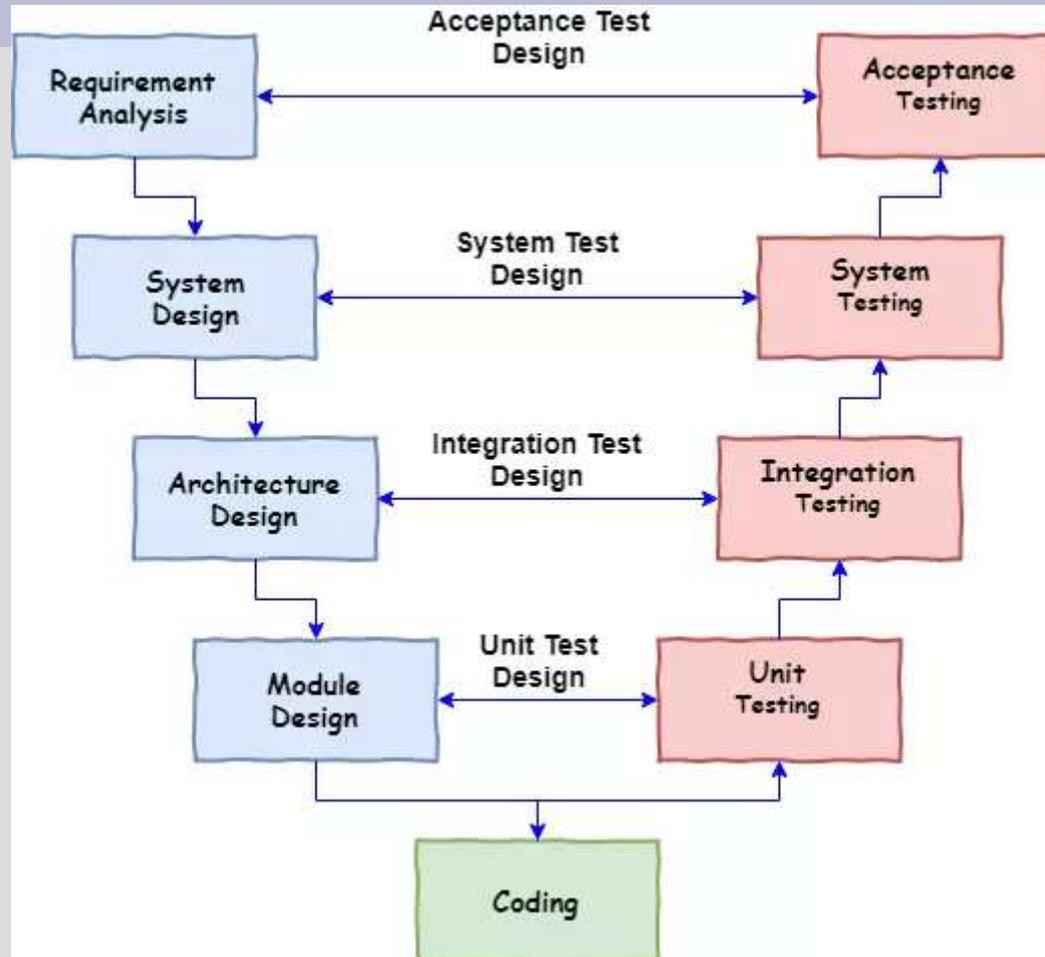
SDLC Iterative Model



In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.

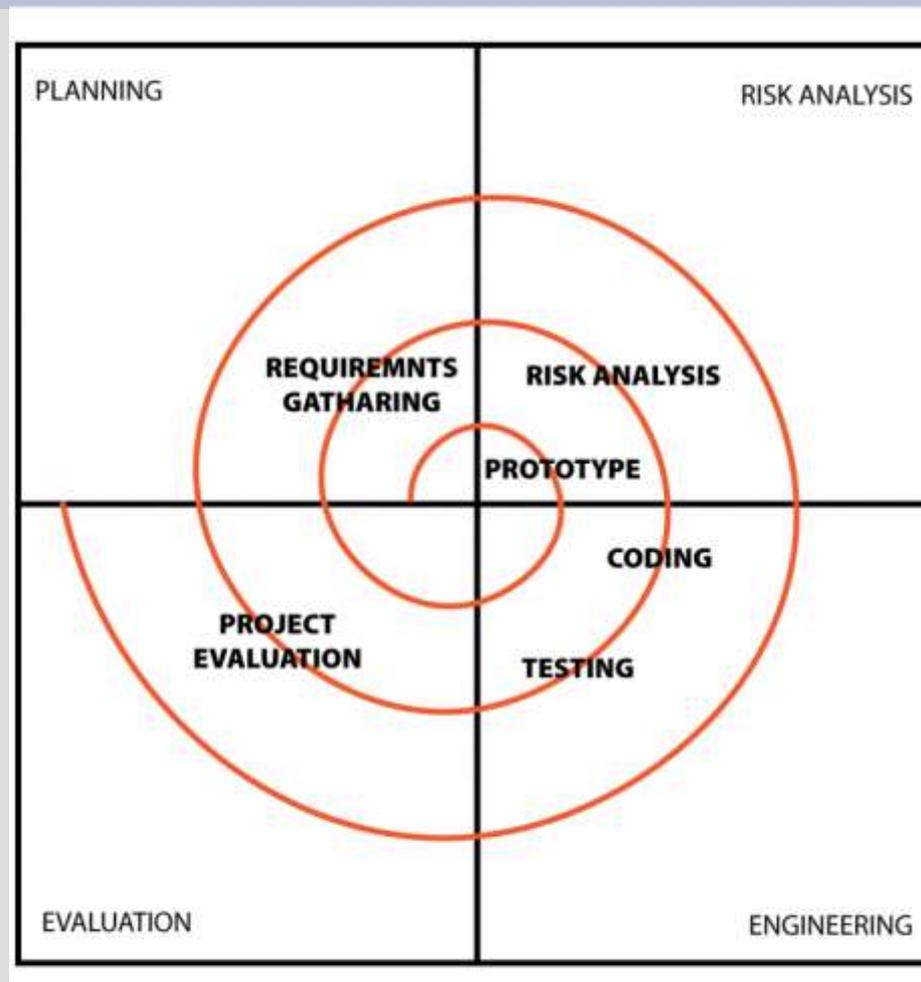
An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

SDLC V-Model



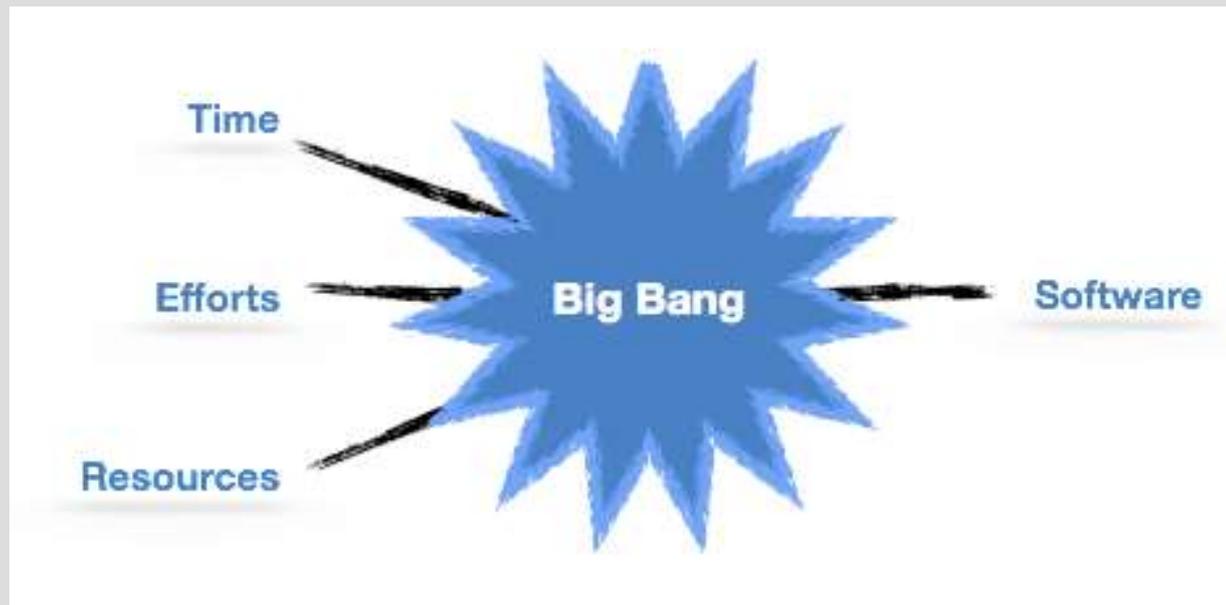
The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.

SDLC Spiral Model



The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spi

SDLC Big Bang Model



The Big Bang model is an SDLC model where we do not follow any specific process. The development just starts with the required money and efforts as the input, and the output is the software developed which may or may not be as per customer requirement. This Big Bang Model does not follow a process/procedure and there is a very little planning required. Even the customer is not sure about what exactly he wants and the requirements are implemented on the fly without much analysis.

Usually this model is followed for small projects where the development teams are very small.

Business Analysis

Business analysis is a research discipline of identifying **business needs** of identifying business needs and determining solutions to **business problems**.

Solutions often include a **software-systems development** component, but may also consist of **process improvement, organizational change** or **strategic planning** and policy development. The person who carries out this task is called a business analyst or BA.

Business analysts do not work solely on developing software systems. But work across the organisation, solving business problems in consultation with business stakeholders. Whilst most of the work that business analysts do today relate to software development/solutions, this derives from the ongoing massive changes businesses all over the world are experiencing in their attempts to digitise.

Although there are different role definitions, depending upon the organization, there does seem to be an area of common ground where most business analysts work. The responsibilities appear to be:

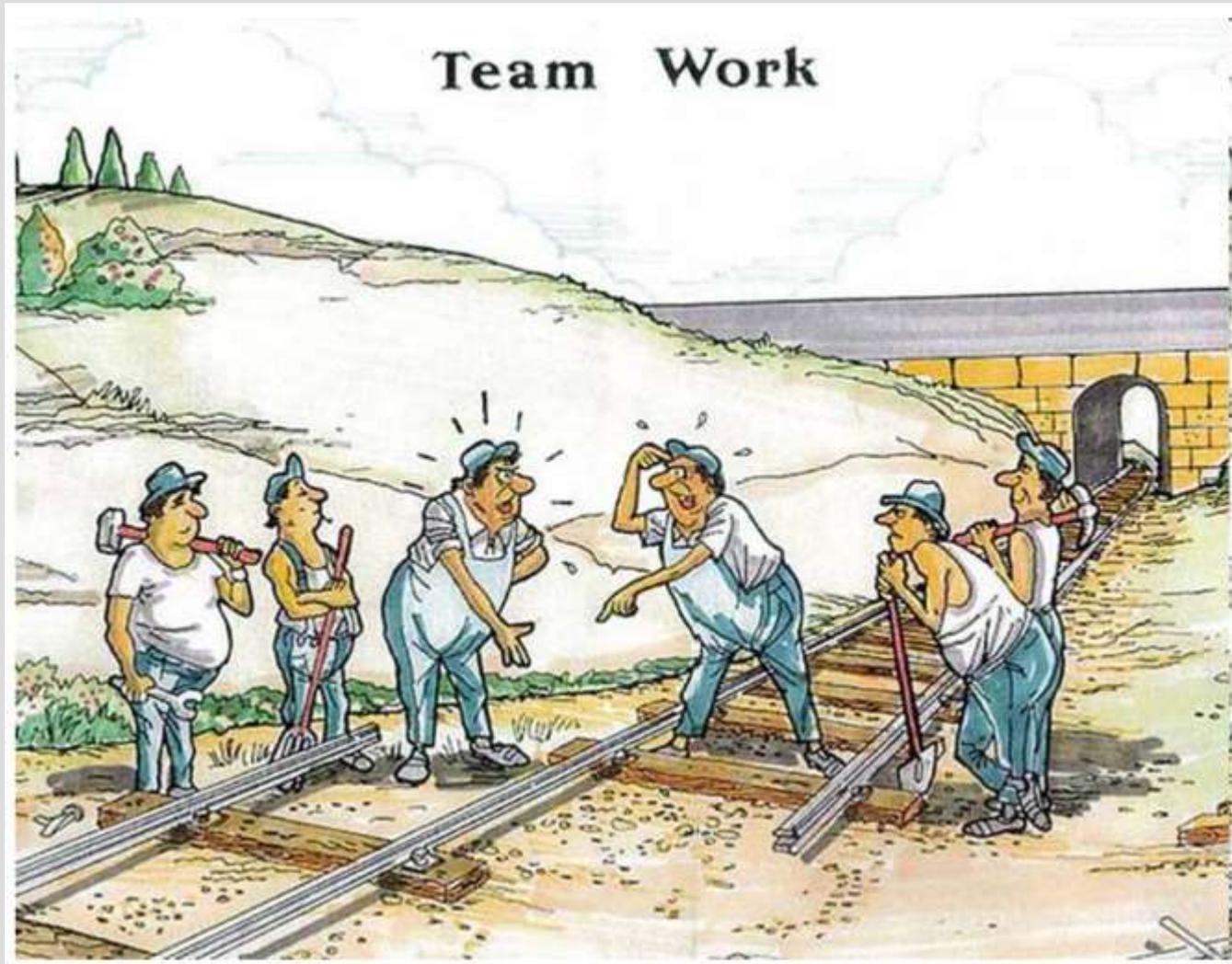
To investigate business systems, taking a holistic view of the situation. This may include examining elements of the organisation structures and staff development issues as well as current processes and IT systems.

To evaluate actions to improve the operation of a business system. Again, this may require an examination of organisational structure and staff development needs, to ensure that they are in line with any proposed process redesign and IT system development.

To document the business requirements for the IT system support using appropriate documentation standards.

Business Analysis

Team Work



Business Analysis

BABOK[®] Guide Knowledge

Business Analysis
Planning and Monitoring

Elicitation and
Collaboration

Requirements Lifecycle
Management

Strategy Analysis

Requirements Analysis
and Design Definition

Solution Evaluation

Business Analysis

Requirement A usable representation of a need.

Need A problem or opportunity to be addressed.

In product development In product development and process optimization, a **requirement** is a singular documented physical or functional need that a particular design, product or process aims to satisfy.

It is commonly used in a formal sense in engineering design is a singular documented physical or functional need that a particular design, product or process aims to satisfy. It is commonly used in a formal sense in engineering design, including for example in systems engineering is a singular documented physical or functional need that a particular design, product or process aims to satisfy. It is commonly used in a formal sense in engineering design, including for example in systems engineering, software engineering is a singular documented physical or functional need that a particular design, product or process aims to satisfy. It is commonly used in a formal sense in engineering design, including for example in systems engineering, software engineering, or enterprise engineering.

It is a broad concept that could speak to any necessary (or sometimes desired) function, attribute, capability, characteristic, or quality of a system for it to have value and utility to a customer, organization, internal user, or other stakeholder. Requirements can come with different levels of specificity; for example, a requirement specification or requirement "spec" (often imprecisely referred to as "the" spec/specs, but there are actually different sorts of specifications) refers to an explicit, highly objective/clear (and often quantitative) requirement (or sometimes, *set* of requirements) to be satisfied by a material, design, product, or service.^[1]

Business Analysis

Requirements clasification

- **Business requirements**
- **Stakeholder requirements**
- **Requirements for the solution**
- *functional*
- *nonfunctional*
- **Requirements for the change process**

Business Analysis

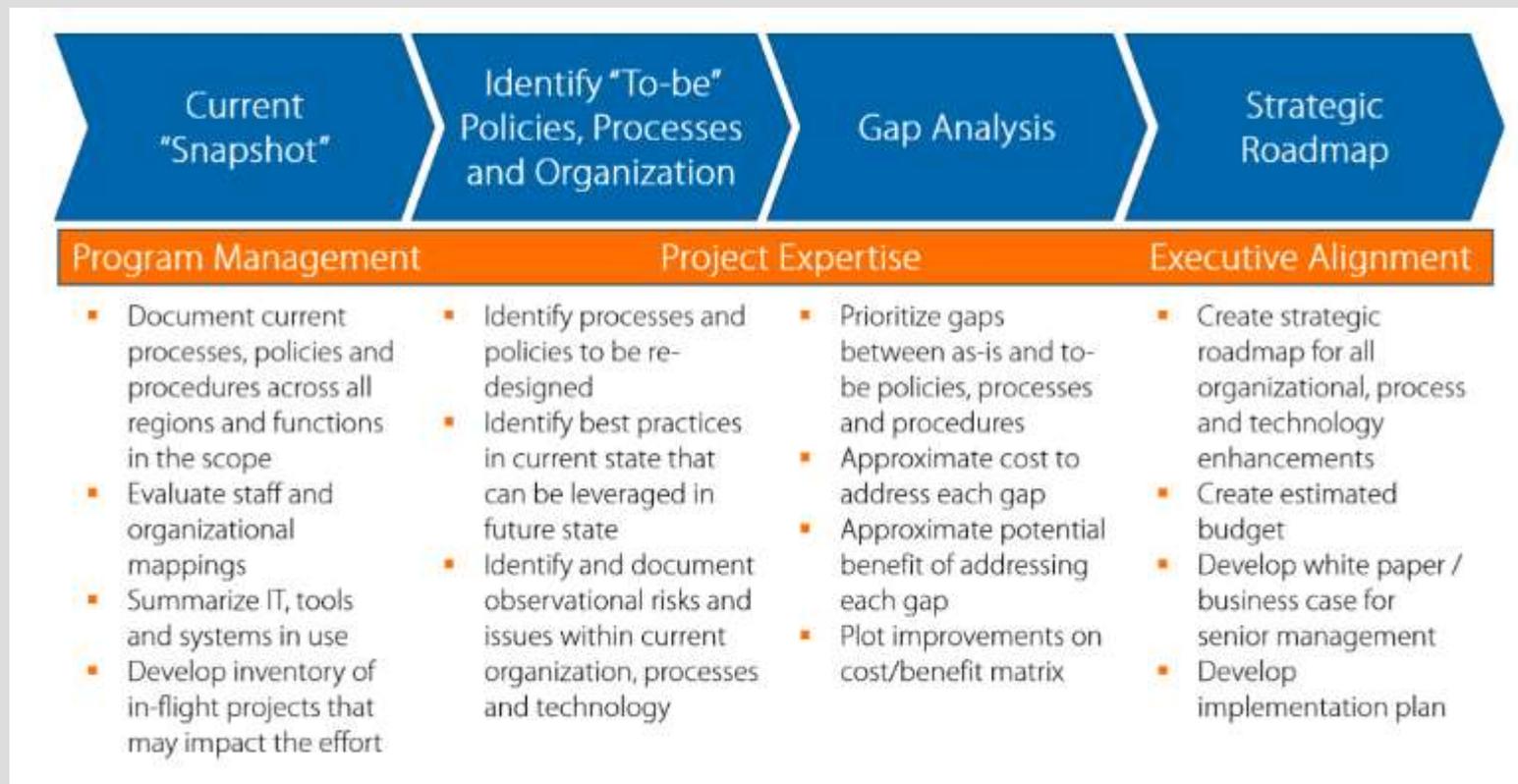
Stakeholder A group or individual with a relationship to the change, the need, or the solution.

Need A problem or opportunity to be addressed.

A stakeholder is a party that has an interest in a company and can either affect or be affected by the business. The primary stakeholders in a typical corporation are its investors, employees, customers and suppliers. However, the modern theory of the idea goes beyond this original notion to include additional stakeholders such as a community, government or trade association.

Business Analysis

Analisis phases



Business Analysis

Regardless of who performs the role, business analysis must be clearly defined and understood as effective business analysis drives project success and business outcomes.

List of key activities that should be considered when defining the definition of business analysis in organization.

1. Analyzing and documenting the business problem and root cause
2. Determining what need to change to fix the problem
3. Evaluating options and recommending the right solution
4. Defining the solution scope
5. Preparing the Business Case
6. Identifying and engaging stakeholders
7. Eliciting business and stakeholder needs
8. Defining requirements
9. Facilitating collaboration between business and development teams
10. Validating and assessing the solution
11. Enabling business change and transformation
12. Ensuring the solution delivers business value

Business Analysis

“Business analysis practitioners include not only people with the job title of business analyst, but may also include:

- o business systems analysts,
- o systems analysts,
- o requirements engineers,
- o process analysts,
- o product managers,
- o product owners,
- o enterprise analysts,
- o business architects,
- o management consultants,

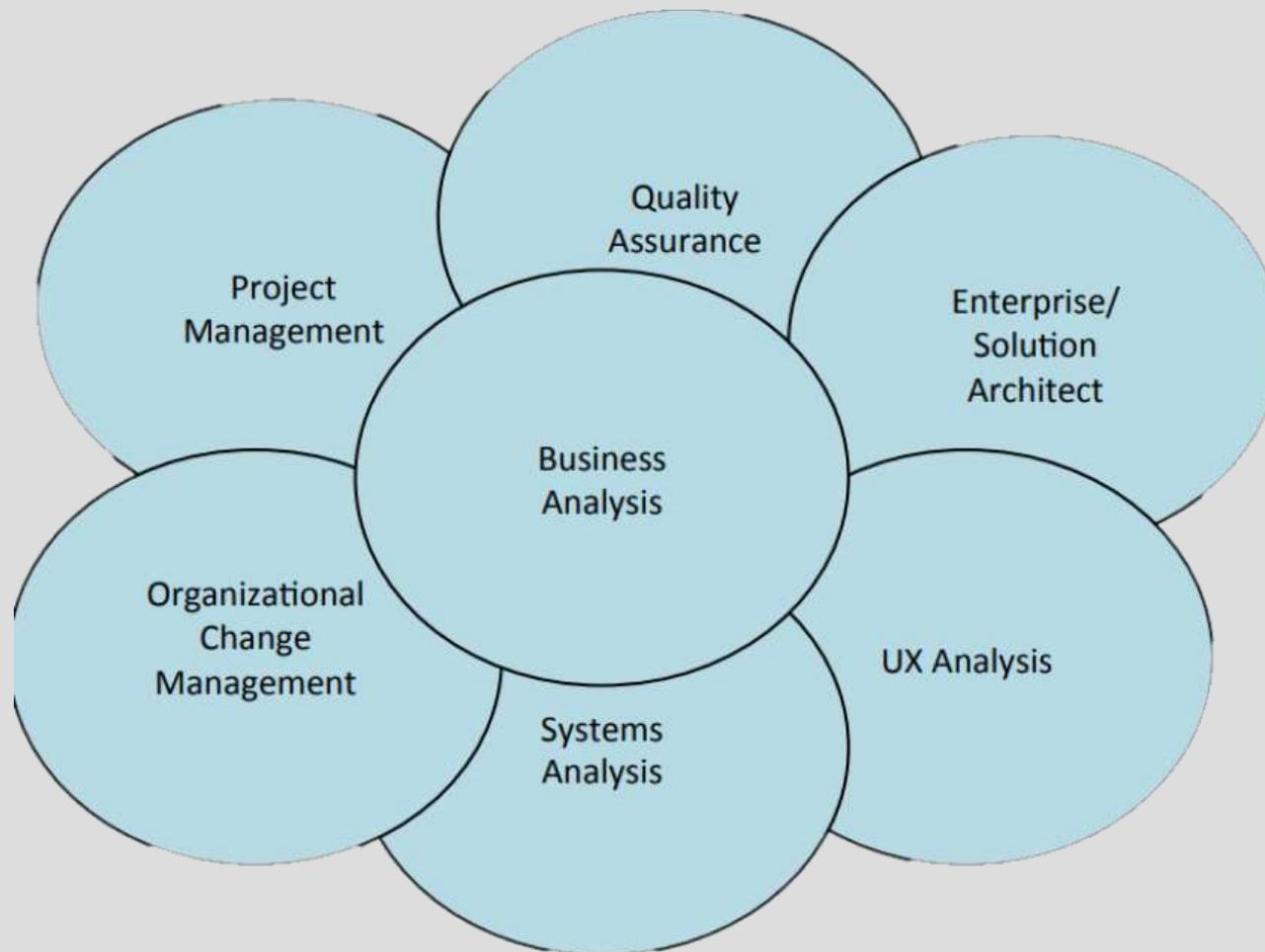
or any other person who performs the tasks described in the BABOK® Guide, including those who also perform related disciplines such as

- o project management,
- o software development,
- o quality assurance, and
- o interaction design.”

Source: BABOK Version 2, IIBA

Business Analysis

BA Role Overlaps Many other Roles



Business Analysis

BABoK?

iIBA™ International Institute
of Business Analysis

*The Guide to the
Business Analysis
Body of Knowledge®*

Version 2.0

Draft for Public Review

BABOK
BABOK
BABOK

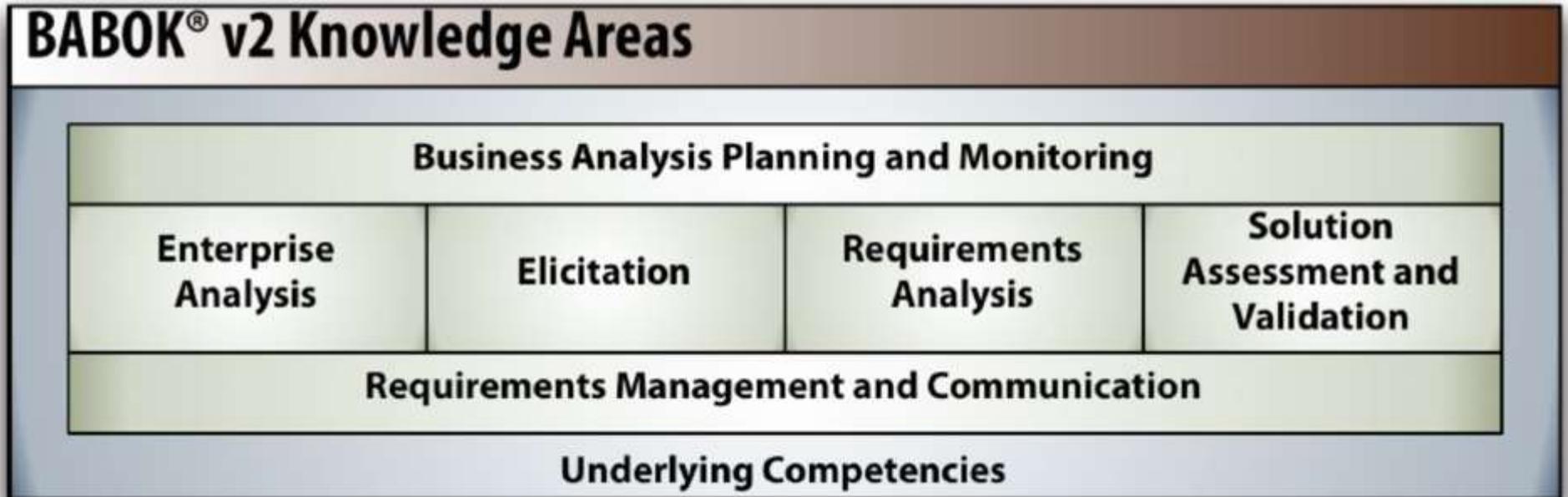
v3

A GUIDE TO THE BUSINESS ANALYSIS
BODY OF KNOWLEDGE®

iIBA™ International Institute
of Business Analysis®

Business Analysis

Areas of business analysis



Business Analysis

Business Analysis Planning and Monitoring:

Describes the tasks used to organize and coordinate business analysis efforts.

Project BA work is a business process with:

- Inputs
- Outputs
- Tasks
- Stakeholders
- Actors

As-Is is our toolbox and methodology

To-be changes with each project

Business Analysis

Business Analysis Planning and Monitoring:

BABOK point of view

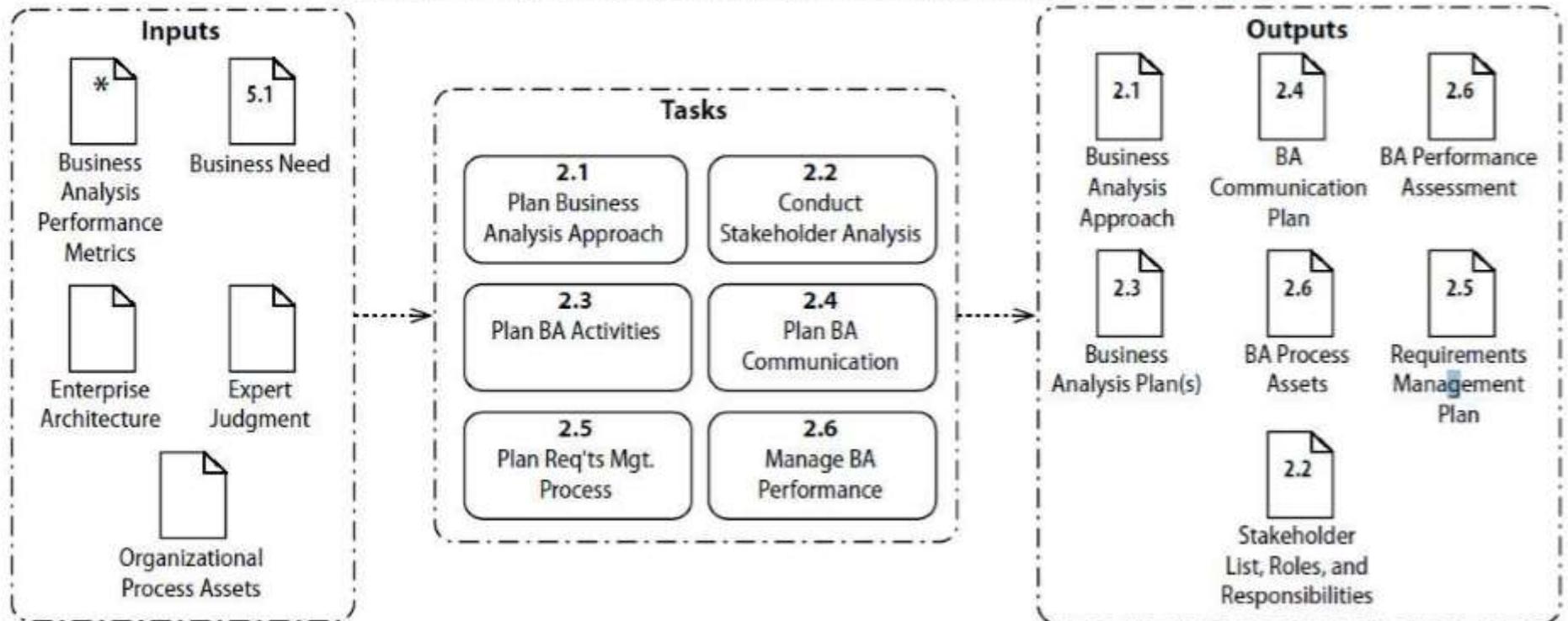
Chapter 2: Business Analysis Planning & Monitoring	17
2.1 Plan Business Analysis Approach	17
2.2 Conduct Stakeholder Analysis	24
2.3 Plan Business Analysis Activities	31
2.4 Plan Business Analysis Communication	37
2.5 Plan Requirements Management Process	42
2.6 Manage Business Analysis Performance	49

Business Analysis

Business Analysis Planning and Monitoring:

BABOK point of view

Figure 2–1: Business Analysis Planning & Monitoring Input/Output Diagram



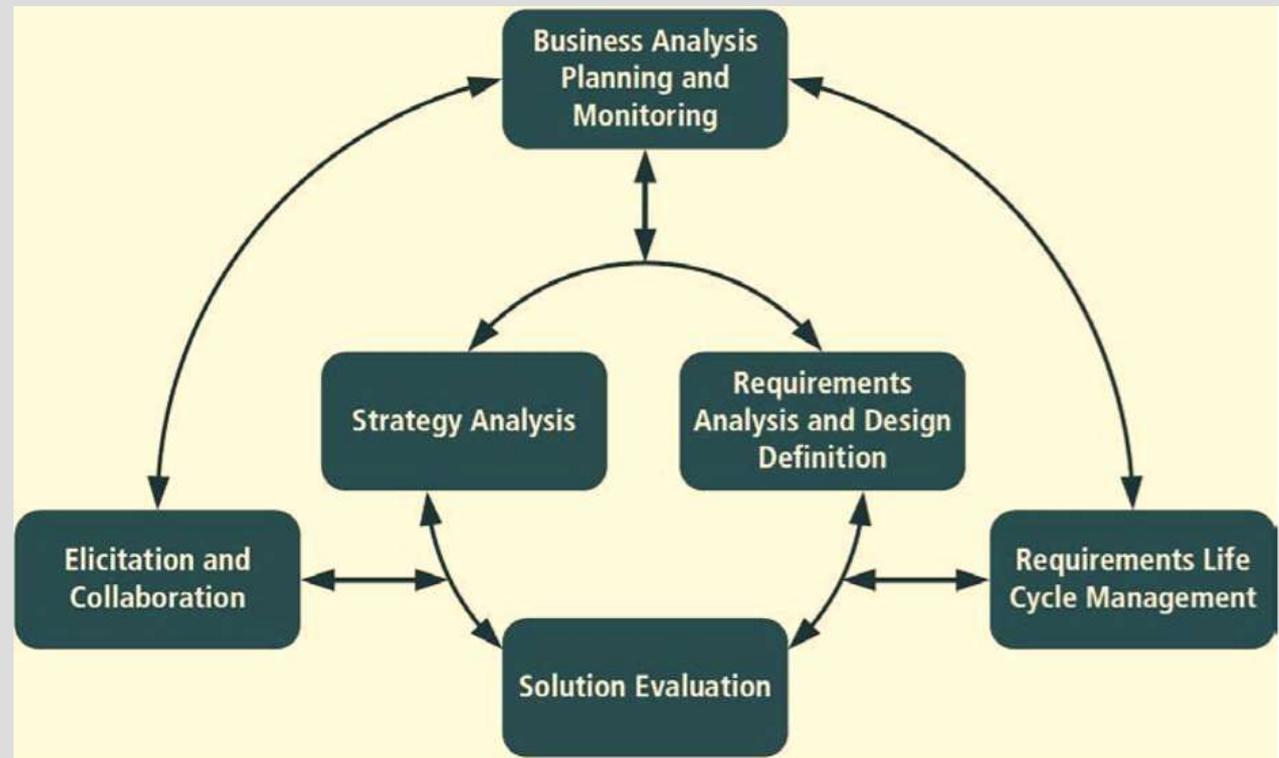
Business Analysis

Elicitation and Collaboration:

Describes the tasks used to prepare for and conduct elicitation activities and confirm the results.

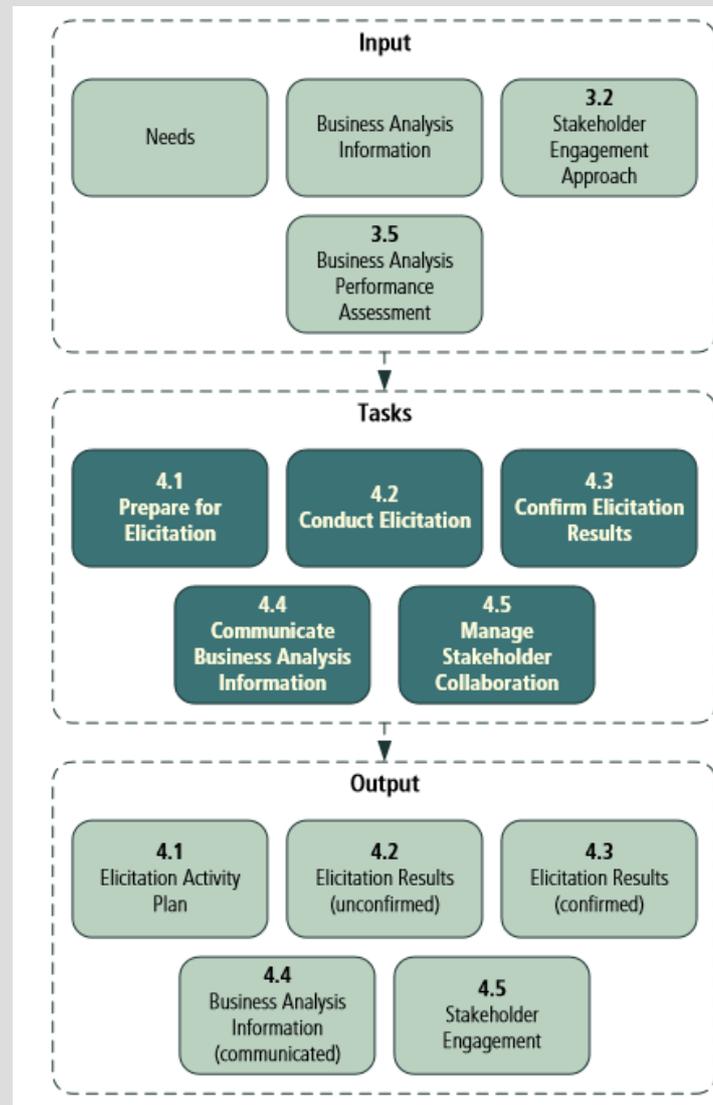
It is collaborative

- It is not a “phase”
- It is ongoing as long as analysis work is occurring
- Should not be considered an “isolated” activity
- May trigger additional elicitation activities to obtain details to fill in gaps and increase understanding



Business Analysis

Elicitation and Collaboration:



Business Analysis

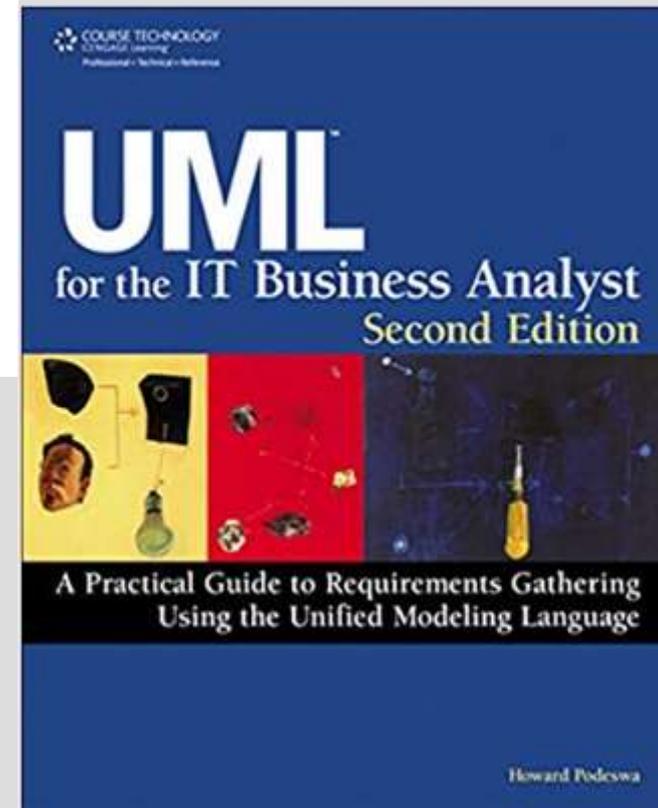
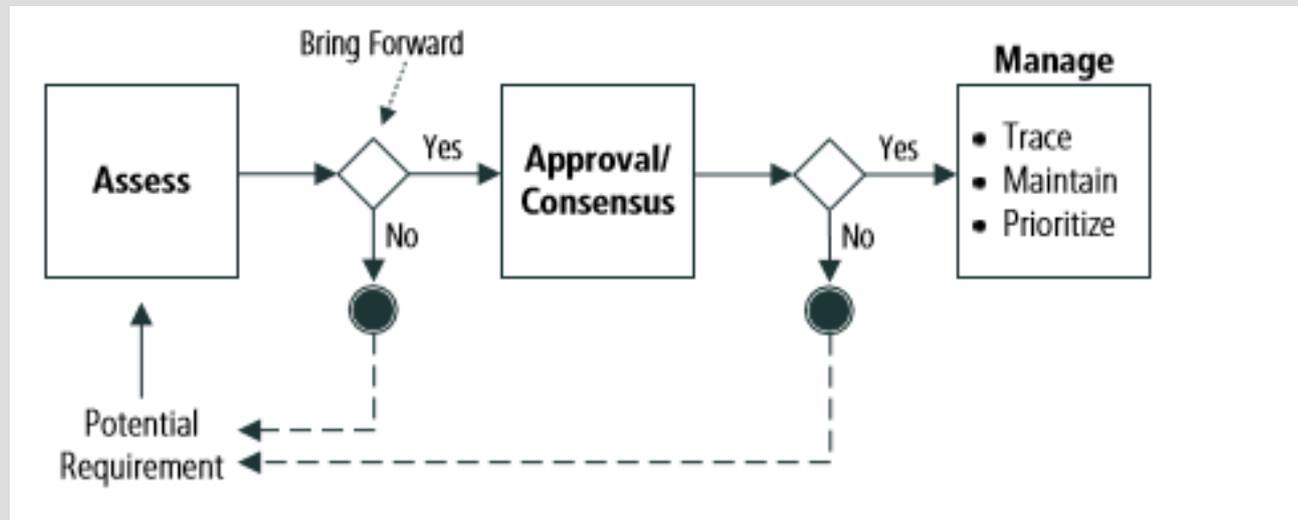
Elicitation and Collaboration:

- **Prepare for Elicitation:** involves ensuring that the stakeholders have the information they need to provide and that they understand the nature of the activities they are going to perform. It also sets a shared set of expectations regarding the outcomes of the activity. Preparation may also involve identifying research sources or preparing to conduct an experiment to see if a process change actually results in an improvement.
- **Conduct Elicitation:** describes the work performed to understand stakeholder needs and identify potential solutions that may meet those needs. This may involve direct interaction with stakeholders, doing research, or running experiments.
- **Confirm Elicitation Results:** involves ensuring that stakeholders have a shared understanding of the outcomes of elicitation, that elicited information is recorded appropriately, and that the business analyst has the information sought from an elicitation activity. This task also involves comparing the information received with other information to look for inconsistencies or gaps.
- **Communicate Business Analysis Information:** provides stakeholders with the information they need, at the time they need it. The information is presented in a useful form, using the right terminology and concepts.
- **Manage Stakeholder Collaboration:** describes working with stakeholders to engage them in the overall business analysis process and to ensure that the business analyst can deliver the outcomes needed.

Business Analysis

Requirements Life Cycle Management:

Describes the tasks used to manage and maintain requirements and design information from inception to retirement.



Business Analysis

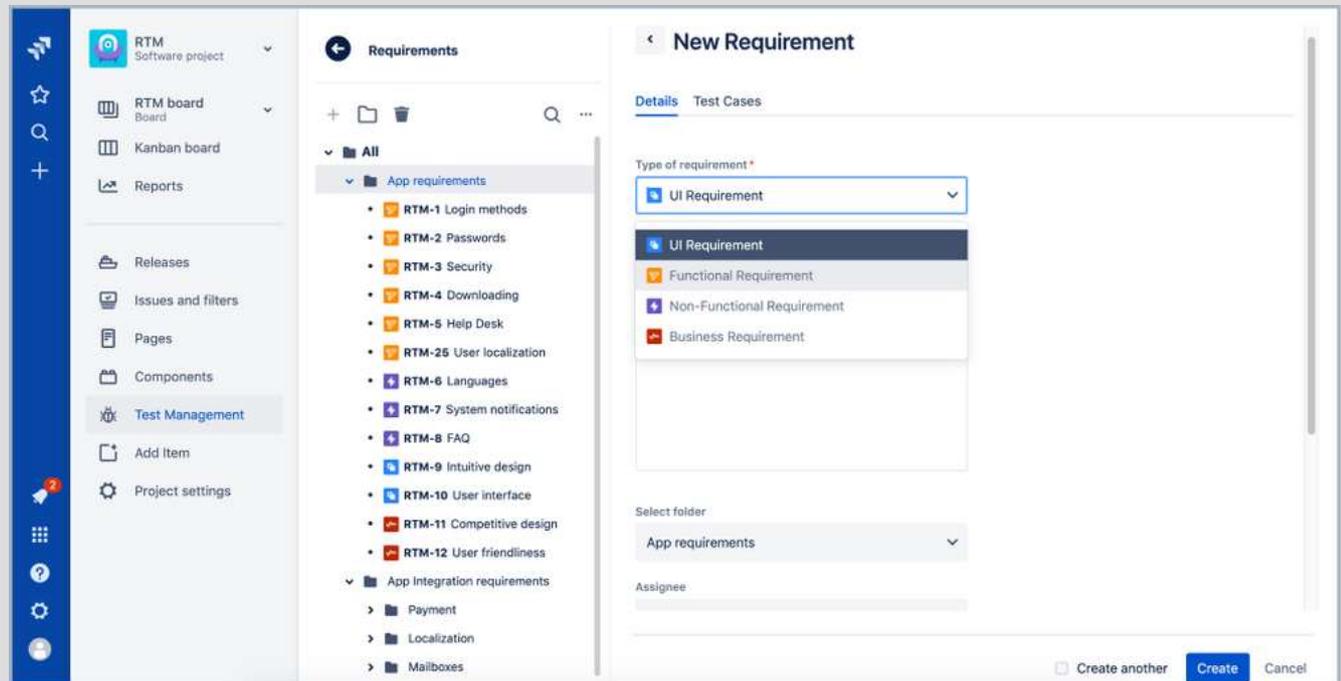
Requirements Life Cycle Management:

- **Trace Requirements:** analyzes and maintains the relationships between requirements, designs, solution components, and other work products for impact analysis, coverage, and allocation.
- **Maintain Requirements:** ensures that requirements and designs are accurate and current throughout the life cycle and facilitates reuse where appropriate.
- **Prioritize Requirements:** assesses the value, urgency, and risks associated with particular requirements and designs to ensure that analysis and/or delivery work is done on the most important ones at any given time.
- **Assess Requirements Changes:** evaluates new and changing stakeholder requirements to determine if they need to be acted on within the scope of a change.
- **Approve Requirements:** works with stakeholders involved in the governance process to reach approval and agreement on requirements and designs.

Business Analysis

Requirements Life Cycle Management:

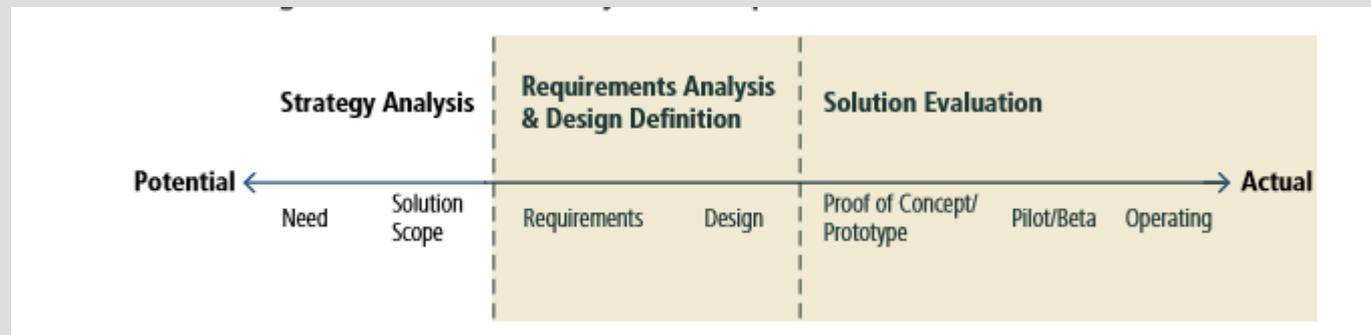
Describes the tasks used to manage and maintain requirements and design information from inception to retirement.



Business Analysis

Enterprise/Strategy Analysis:

Describes the tasks used to identify the business need, address that need, and align the change strategy within the enterprise.



Business Analysis

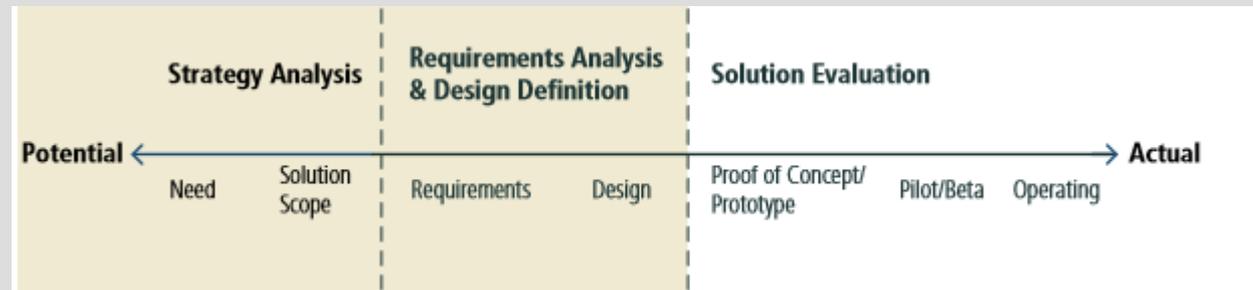
Enterprise/Strategy Analysis:

- **Analyze Current State:** understands the business need and how it relates to the way the enterprise functions today. Sets a baseline and context for change.
- **Define Future State:** defines goals and objectives that will demonstrate that the business need has been satisfied and defines what parts of the enterprise need to change in order to meet those goals and objectives.
- **Assess Risks:** understands the uncertainties around the change, considers the effect those uncertainties may have on the ability to deliver value through a change, and recommends actions to address risks where appropriate.
- **Define Change Strategy:** performs a gap analysis between current and future state, assesses options for achieving the future state, and recommends the highest value approach for reaching the future state including any transition states that may be required along the way.

Business Analysis

Solution Evaluation:

Describes the tasks used to assess the performance of and value delivered by a solution and to recommend improvements on increasing values.

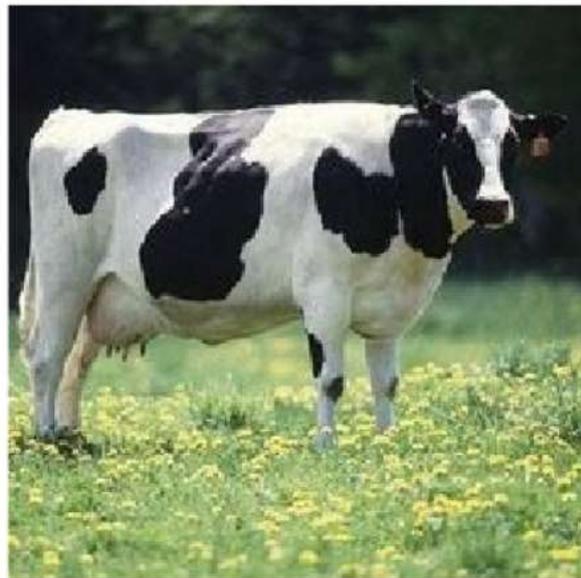


Business Analysis

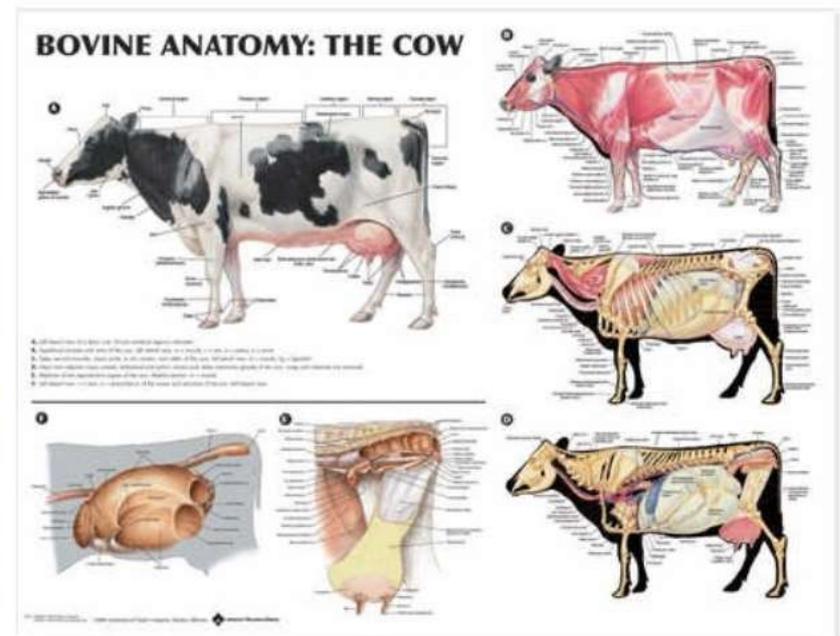
Analysis

Describes the tasks used to identify the business need, address that need, and align the change strategy within the enterprise.

Business Design



Enterprise Architecture



Analysis

What is UML

- Unified modeling language (UML) for visualizing, specifying, constructing, documenting of artifact of a software system
- The blueprint of a system is written in it UML is also used for modeling non-software system
- It is standard for building object oriented and component based software system
- UML is a notation system through which we can visualize a model of a system
- It describe only design or structure of system

Analysis

History of UML

- In 1997 OMG (Object management group) a non profit organization standardized the UML
- Since then they are making improvement in UML
- Released versions of UML

Analysis

History of UML

- Versions
- 2.2 February 2009 <http://www.omg.org/spec/UML/2.2>
- 2.1.2 November 2007 <http://www.omg.org/spec/UML/2.1.2>
- 2.1.1 August 2007 <http://www.omg.org/spec/UML/2.1.1>
- 2.0 July 2005 <http://www.omg.org/spec/UML/2.0>
- 1.5 March 2003 <http://www.omg.org/spec/UML/1.5>
- 1.4.2 July 2004 ISO/IEC 19501
- 1.4 September 2001 <http://www.omg.org/spec/UML/1.4>
- 1.3 March 2000 <http://www.omg.org/spec/UML/1.3>
- 1.2 July 1998
- 1.1 November 1997

Analysis

Importance of Software Model

- If software is complex need of model
- If a system is developed by the hundreds of People
- there is also need of model for smooth communication and understanding the software system
- manage complexity in design
- build and design architecture
- Visualize the implementation
- Design secure , scalable, robust and extendable system

Analysis

UML 2.0 Diagrams

- UML is divided into two General set of Diagrams
- Structured modeling diagrams It depicts the static view of the model
- Behavioral modeling diagram Behavior diagrams depicts the varieties of interaction within a model as it 'executes' over time

Analysis

Structural Modeling Diagram

- Class Diagram
- Object Diagram
- Component Diagram
- Package Diagram
- Composite Structure
- Deployment Diagram

Analysis

Behavioral Modeling Diagram

- Use Case Diagram
- Activity Diagram
- State Machine Diagram
- Communication Diagram
- Sequence Diagram
- Timing Diagram
- Interaction Over view Diagram

Analysis

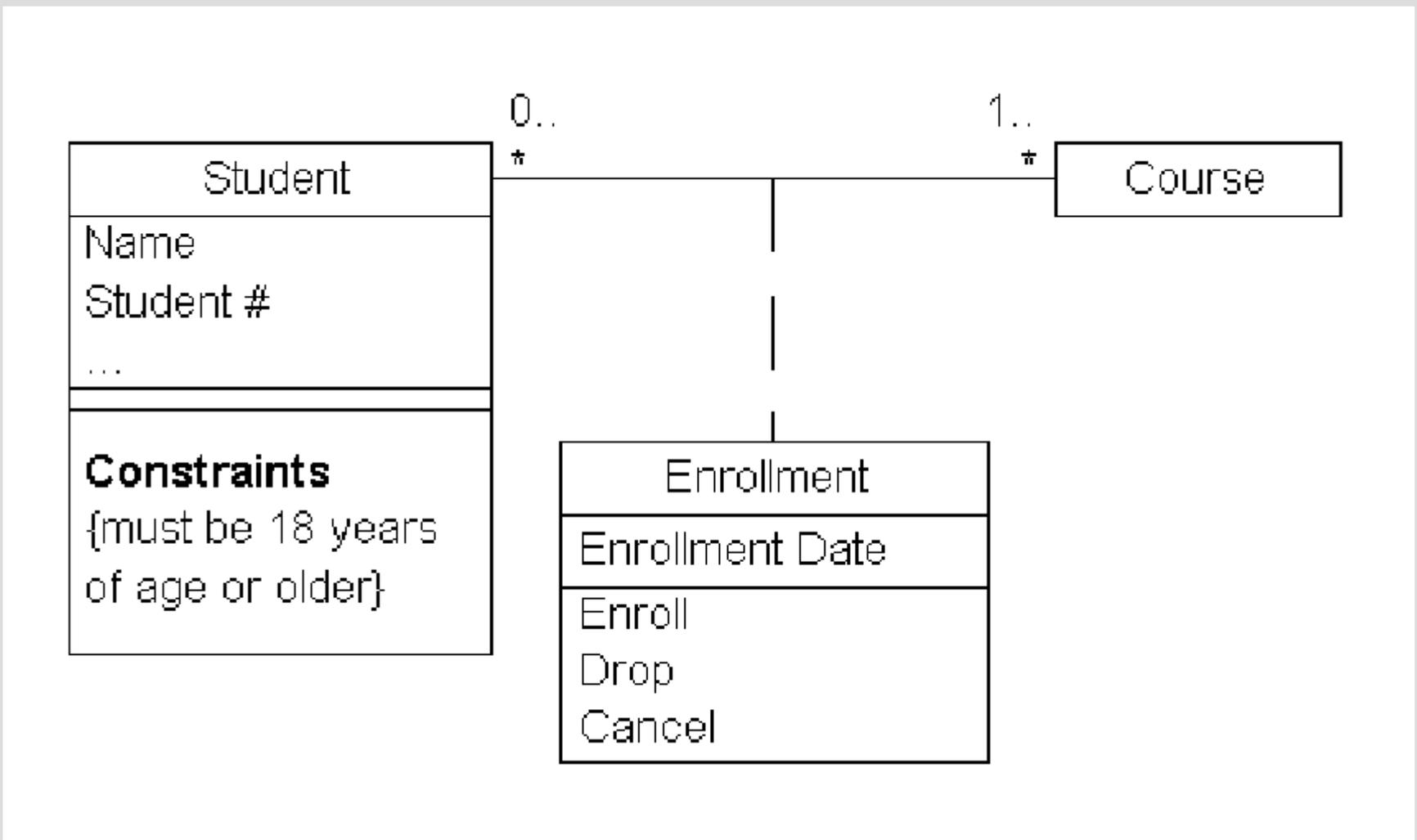
Class diagram

UML class diagrams describe the classes of the system, their inter-relationships, and the operations and attributes of the classes

- Explore domain concepts in the form of a domain model
- Analyze requirements in the form of a conceptual/analysis model
- Depict the detailed design of object-oriented or object based software

Analysis

Class diagram



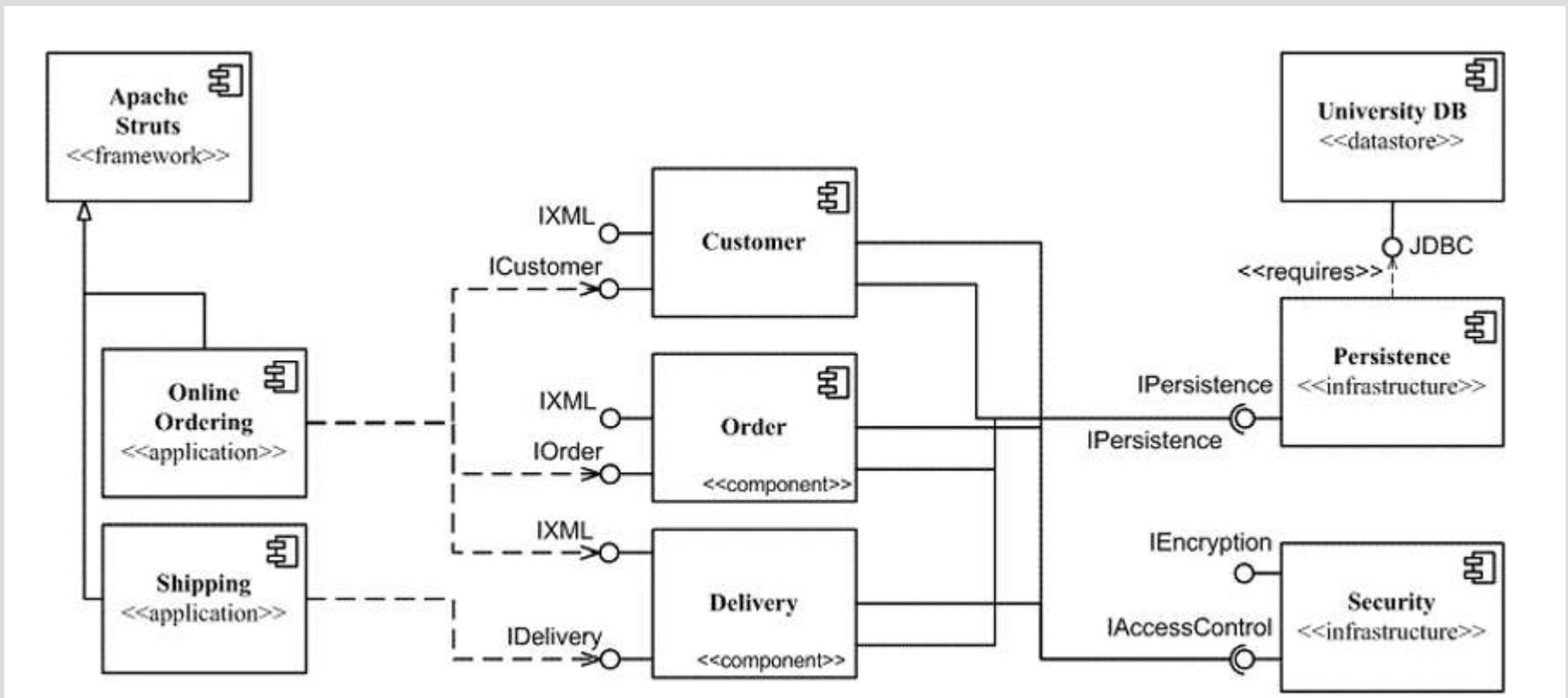
Analysis

Component diagram

UML component diagrams shows the dependencies among software components, including the classifiers that specify them (for example implementation classes) and the artifacts that implement them; such as source code files, binary code files, executable files, scripts and tables.

Analysis

Component diagram



Analysis

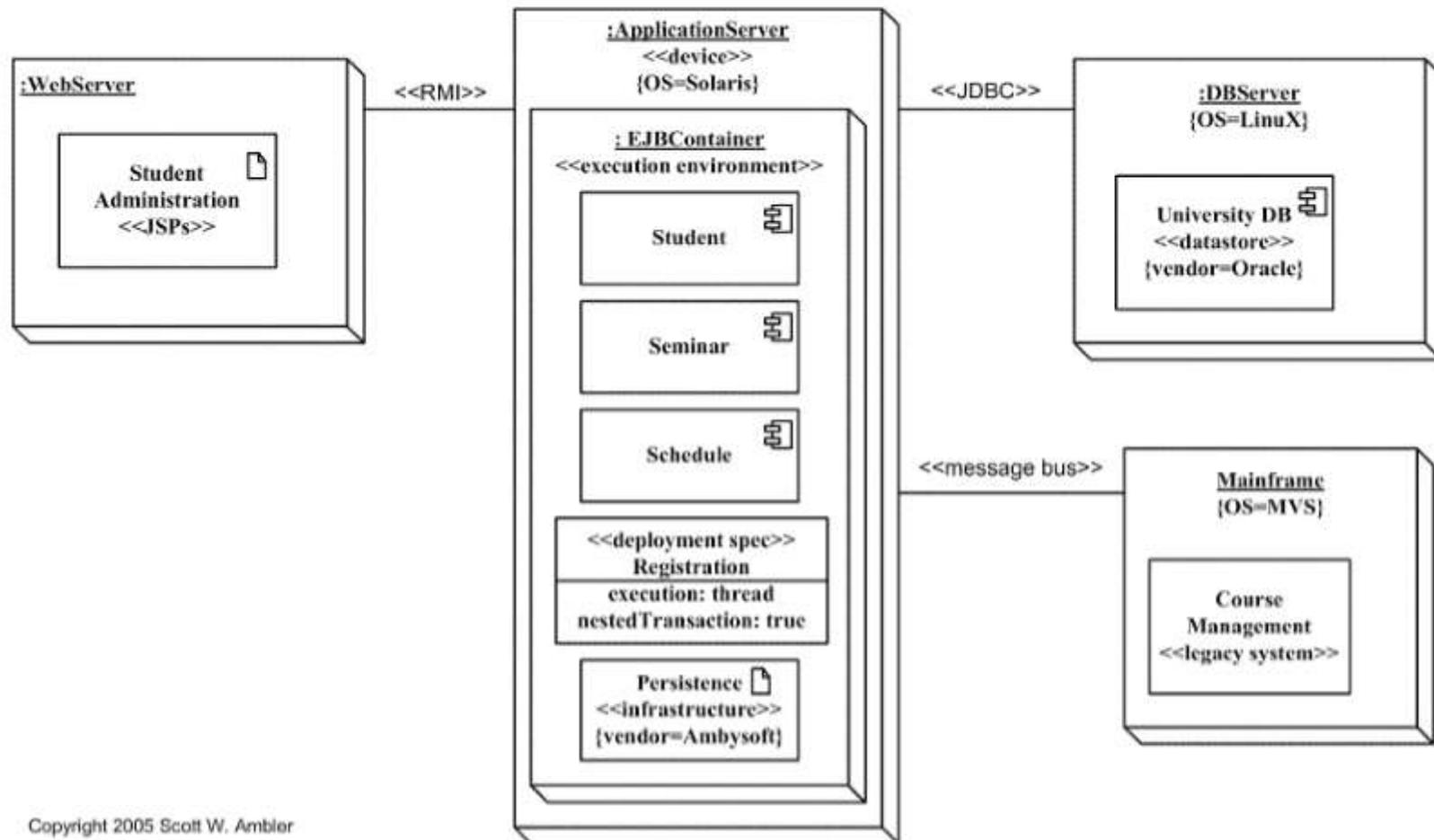
Deployment diagram

UML deployment diagram depicts a static view of the run-time configuration of hardware nodes and the software components that run on those nodes.

Deployment diagrams show the hardware for your system, the software that is installed on that hardware, and the middleware used to connect the disparate machines to one another.

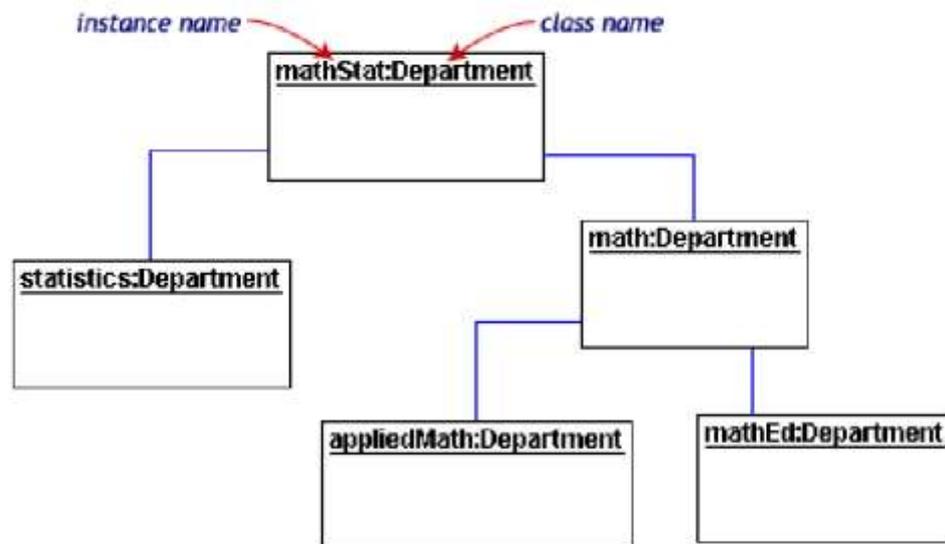
Analysis

Deployment diagram

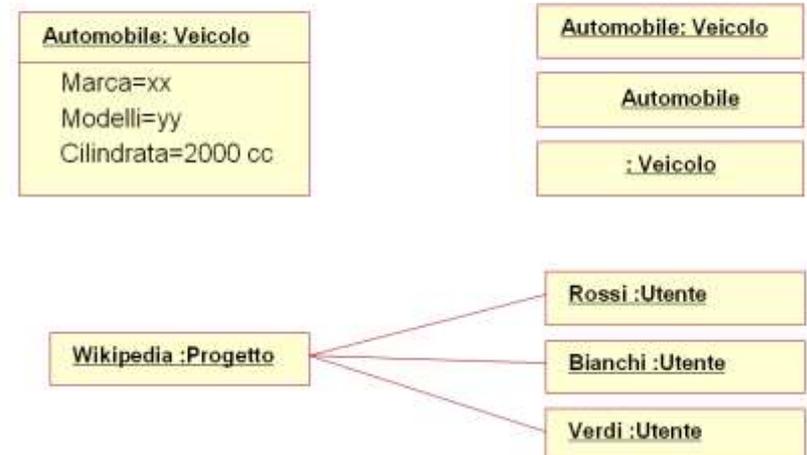


Analysis

- **UML 2 Object diagrams** (instance diagrams), are useful for exploring real world examples of objects and the relationships between them. It shows instances instead of classes. They are useful for explaining small pieces with complicated relationships, especially recursive relationships.



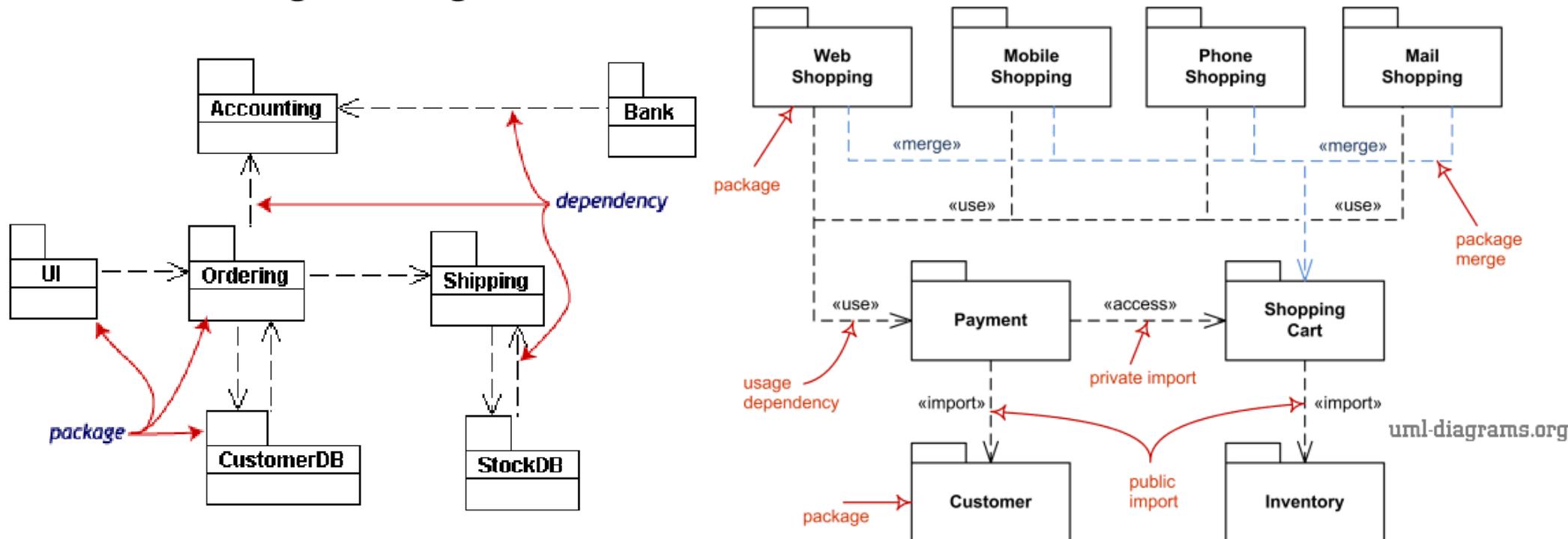
Object Diagram



Analysis

Package diagram

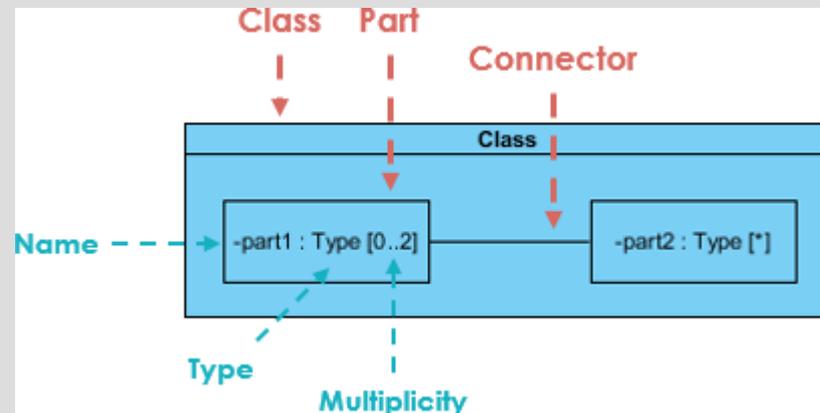
- UML 2 Package diagrams simplify complex class diagrams, it can group classes into packages. A package is a collection of logically related UML elements. Packages are depicted as file folders and can be used on any of the UML diagrams.



Analysis

Composite structure diagram

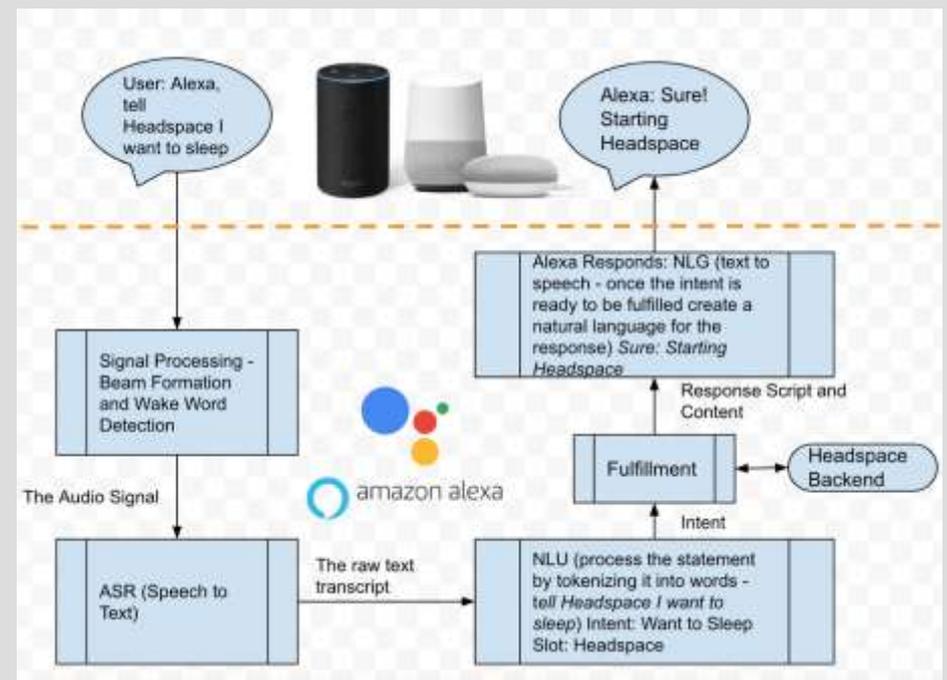
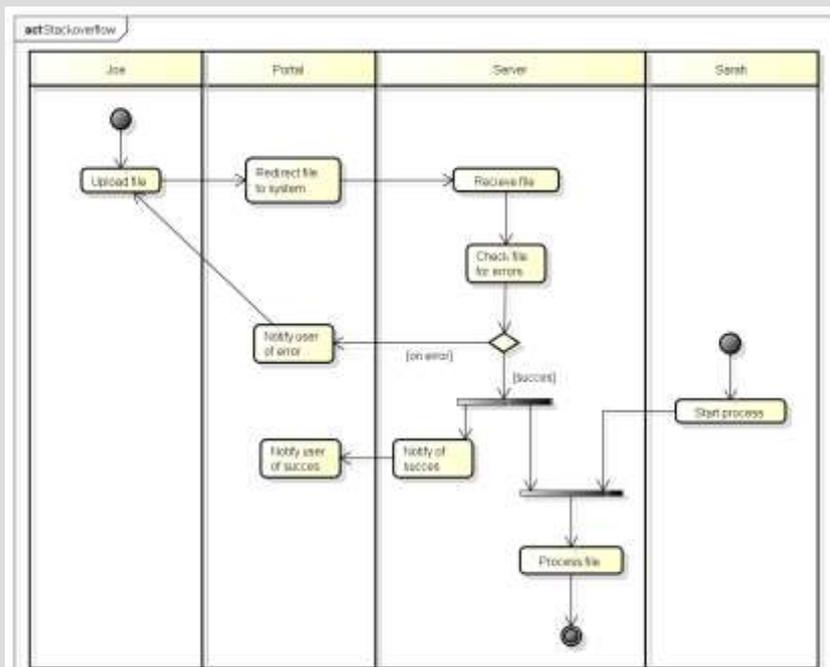
- UML 2 Composite structure diagrams used to explore run-time instances of interconnected instances collaborating over communications links. It shows the internal structure (including parts and connectors) of a structured classifier or collaboration.



Analysis

Activity diagram

- UML 2 Activity diagrams helps to describe the flow of control of the target system, such as the exploring complex business rules and operations, describing the use case also the business process. It is object-oriented equivalent of flow charts and data-flow diagrams (DFDs).



Analysis

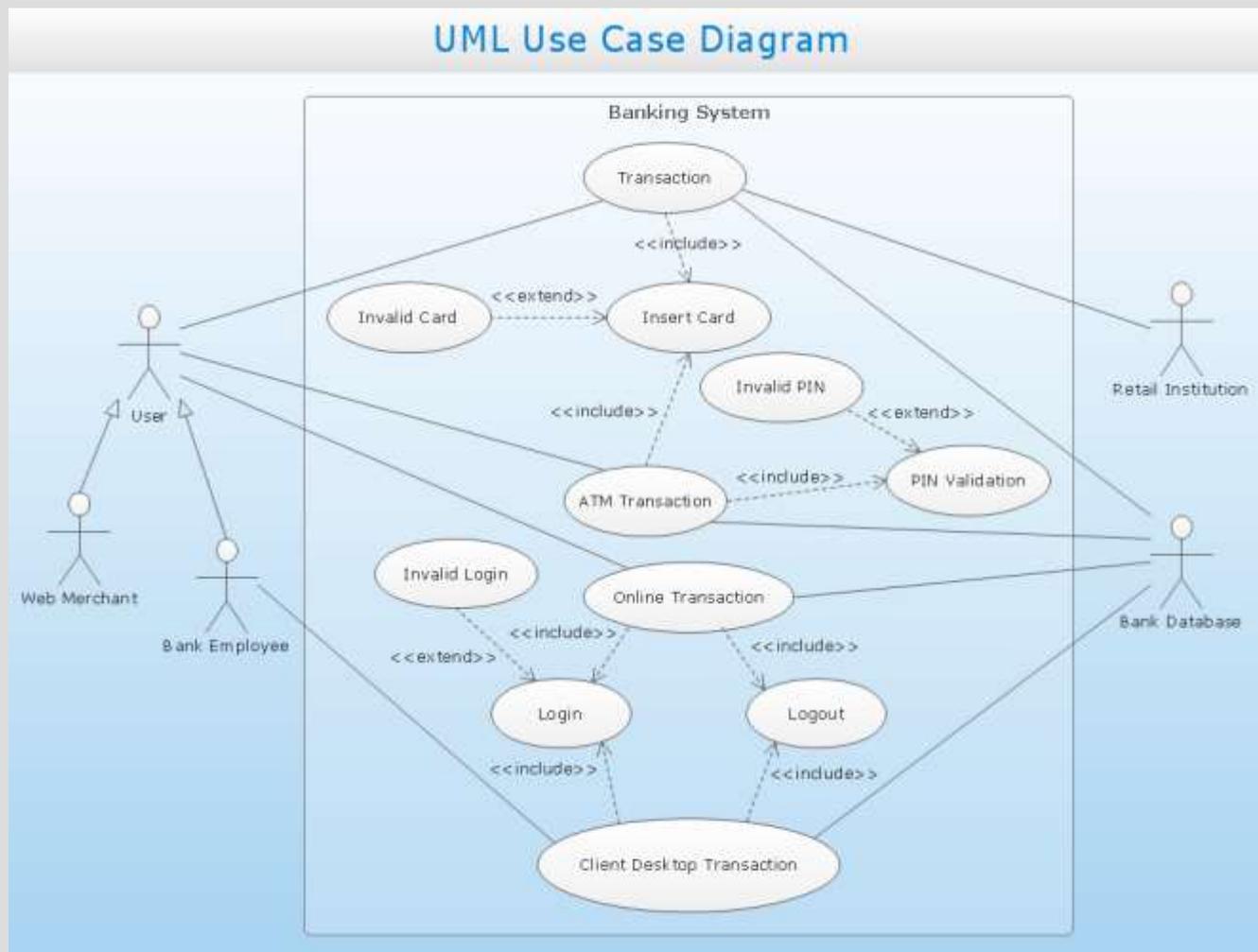
Use cases diagram

UML 2 Use cases diagrams describes the behavior of the target system from an external point of view. Use cases describe "the meat" of the actual requirements.

- Use cases. A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- Actors. An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.
- Associations. Associations between actors and use cases are indicated by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.

Analysis

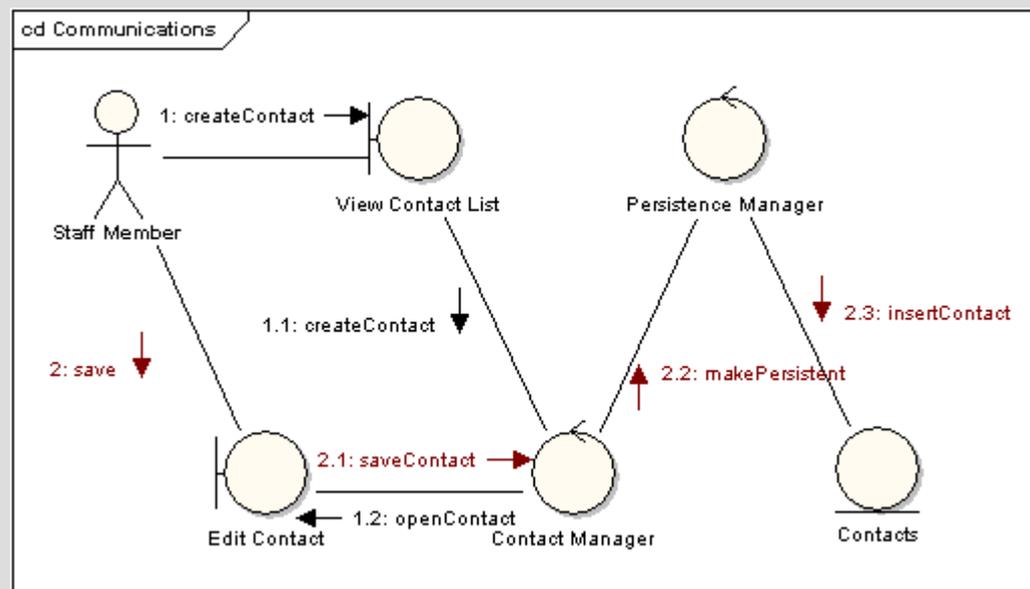
Use cases diagram



Analysis

Communication diagram

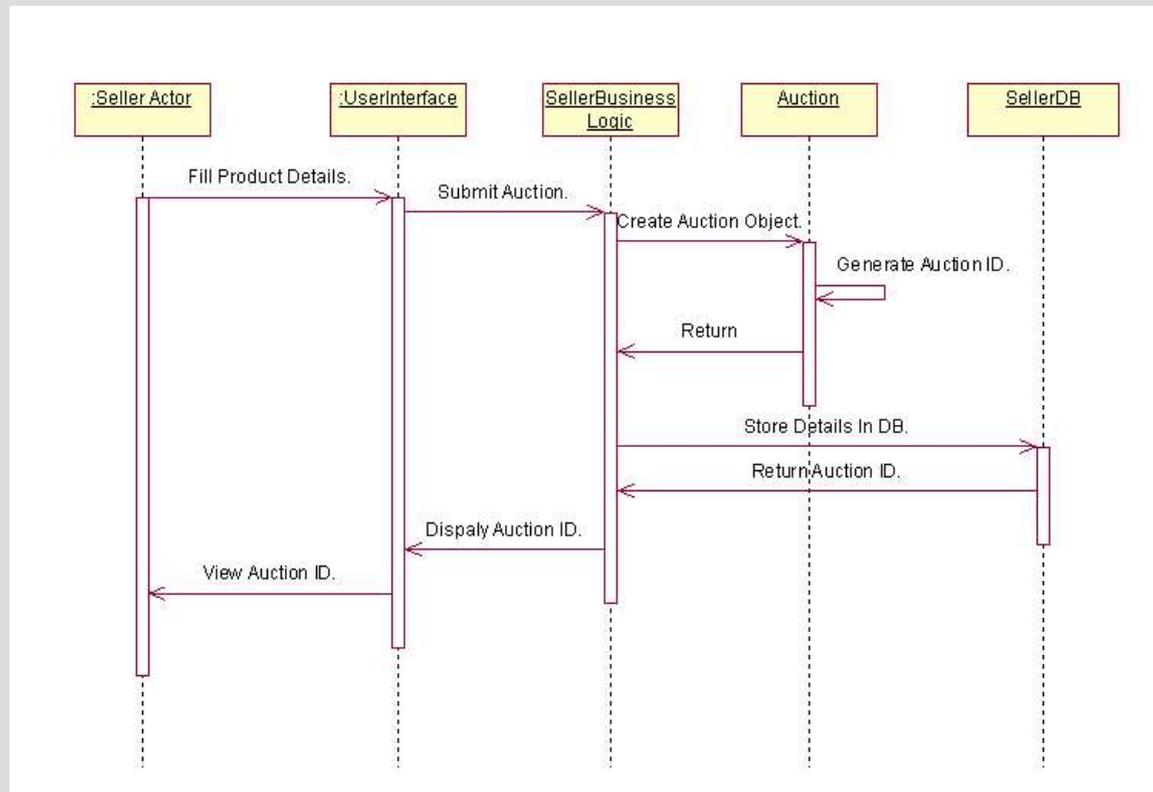
- UML 2 Communication diagrams used to model the dynamic behavior of the use case. When compare to Sequence Diagram, the Communication Diagram is more focused on showing the collaboration of objects rather than the time sequence.



Analysis

Sequence diagram

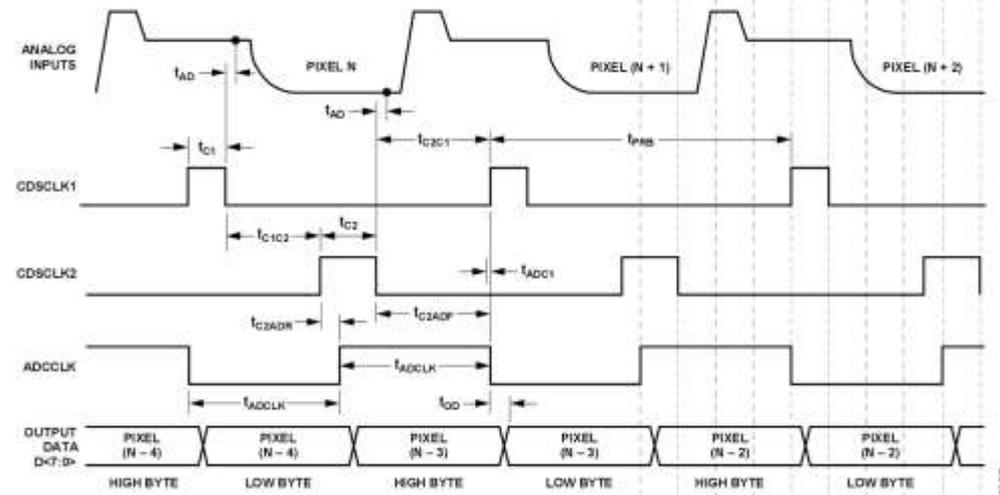
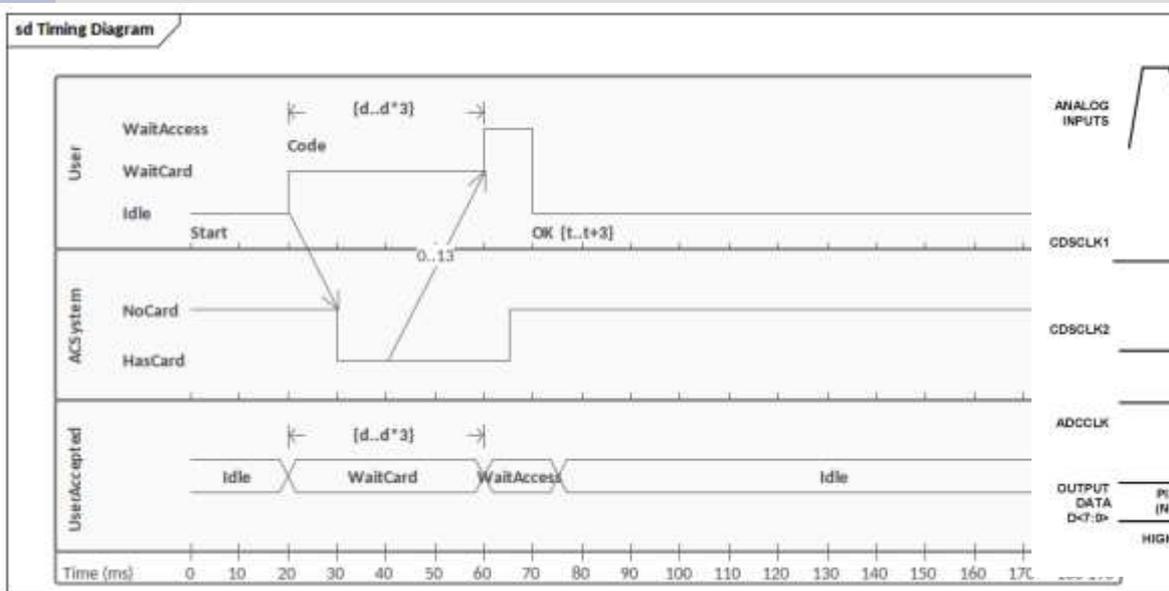
- UML 2 Sequence diagrams models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case.



Analysis

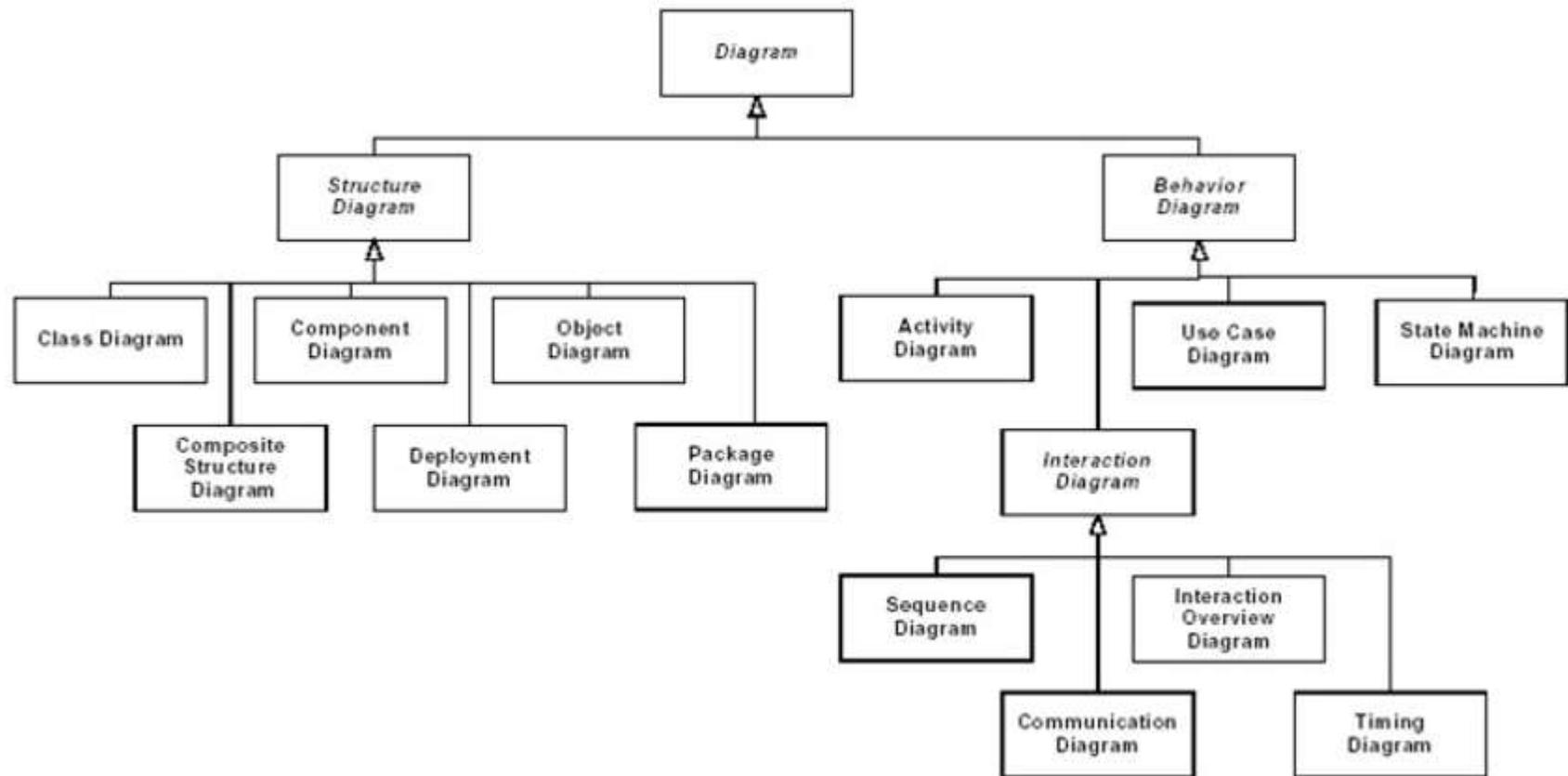
Timing diagram

- UML 2 Timing diagrams shows the behavior of the objects in a given period of time. Timing diagram is a special form of a sequence diagram. The differences between timing diagram and sequence diagram are the axes are reversed so that the time are increase from left to right and the lifelines are shown in separate compartments arranged vertically



Analysis

UML diagram hierarchy



Analysis

References

- <http://www.agilemodeling.com/>
- <http://www.visualparadigm.com/VPGallery/diagrams/index.html>
- <http://bdn.borland.com/article/0,1410,31863,00.html>
- http://en.wikipedia.org/wiki/Unified_Modeling_Language
- http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/index.htm

IT Architecture

What is IT architecture

Technical foundation of an effective IT strategy
- core of successful modern business strategy

Technology plan for managing IT Investment

- Defines the components of overall information system
- Provides plan from to procure/develop systems that will work together
- Enables to manage IT investment in way that meet the needs of business