

• Rest (ROA: Resource Oriented Architecture)

Rest (Representational state transfer) is a sw architectural style for communicating distributed system. the REST architecture is based on Http. It works using a well-defined URL structure that uniquely identifies a resource or a set of resources, and using specific HTTP methods for retrieving or modifying information (GET, POST, PUT, DELETE).

The term "Representational State Transfer" means that a server will respond with the representation of a resource in HTML, XML or JSON document format.

Moreover, Rest is also stateless, meaning that each request is independent and contains all the necessary information.

• SOAP (SOA: Service Oriented Architecture)

SOAP is a standard protocol for exchanging information between application in a distributed environment, using XML as the message format.

Key features of SOAP:

- Message Format: SOAP messages are encoded only in XML
- Transport Protocol: SOAP can be transported over various protocols
- Platform Independence

A Soap Message consist of three main parts:

1. Envelope: it is the main container of the message
2. Header (Optional): Contain control information
3. Body: Contains the main data of the message, whether it is a request or a response.

Soap can be implemented in two ways for exchanging data and invoking services between a client and a server:

- Remote Procedure Call (RPC): Allows an application to invoke methods on a remote server. The soap message contains the name of the procedure to execute and its parameters. The server processes the request and returns the result in the body of the SOAP response.
- Document Oriented: The client sends an XML message / Document to the server. The server processes the document and generates a response.

• Function Point

FP are a measurement technique to evaluate the complexity and size of a SW system based on the functional provided to users. They represent a measurement independent of the technology and programming language used.

The main components are:

- EI (external input): data or information entered by the user
- EO (external output): data or information returned to the user
- EQ (external query): data request answered immediately
- ILF (Internal logical File): data stores managed by the application
- EIF (External Interface File): Referenced but unmanaged data stores

Initially each component is classified as simple, medium or complex (using DET, RET, FTR) and then, consulting the cost table, each one is assigned the relative FP value.

• CoCoMo

Cocomo is a effort/cost estimation model for SW development, it is based on empirical and quantitative parameters such as SW size and various factors that influence development.

The effort is calculated in Person-Months (PM) using

$$PM = A \times (\text{Size})^E \times \prod AF_i$$

$$E = B + 0.01 \left(\sum_{i=1}^5 SF \right)$$

Where AF are the adjustment factors and SF are the scaling factors (Precedentedness, Team cohesion, Risk Resolution . . .)

While the development time is calculated as :

$$TDEV = C \times (PM)^F \times \frac{SCED}{100}$$

$$F = D + 0.2(E - B)$$

In Cocomo II there are two main models:

- Early Design Model: useful for obtaining estimates during the development analysis phase. It's applied when the user requirements have been defined. 7 Adjustment Factors
- Post-Architecture Model: It's used when the project is finished. 17 Adjustment Factors are used

FP focus on the functional complexity of the SW, and Cocomo translates this complexity into estimated cost, effort and duration, considering additional factors.

• Scrum

Scrum is an Agile framework used in software development that allows us to focus on delivering the highest business value in the shortest time.

It is based on :

- short and frequent iterations (2-4 weeks) called sprints
- a continuous focus on the highest priority features
- self-organized teams that define how work gets done
- the ability to release working sw at the end of each sprint

Sprint Framework

Roles

- Product owner : defines and prioritizes features, accept or rejects the sprint results
- Scrum Master : Organize and protect the team from the external environment
- Team : consisting of 5-9 members

Ceremonies

- Sprint Planning : Sprint activity planning
- Daily Scrum : Daily 15 minute meeting to synchronize the team
- Sprint review : Presentation of completed work in presence of the stakeholders
- Sprint retrospective : At the end of each sprint there is an evaluation of what worked and what to improve

Artifacts

- Product Backlog : Prioritized list of all features
- Sprint Backlog : Goal of the sprints and the estimated hours to complete them
- Sprint burndown chart : Sprint progress graph

• DevOps

DevOps is a methodology that combines sw developments and IT operations (Release, configure and monitor)
It aims to improve collaboration, automation and speed across the entire sw lifecycle.

Key definition of DevOps.

- Collaboration and Communication: DevOps brings together developers and operations team to share responsibility and improve transparency
- Automation: Delivery and Infrastructure change are automated to reduce time and errors.
- Continuous Learning and Improvement: DevOps promote a mindset of constant growth, viewing mistakes as learning opportunities

DevOps transforms the traditional sw delivery cycle into a continuous cycle, integrating:

- Continuous Integration (CI): Developers integrate the code into a shared repository. (Github)
- Continuous Delivery (CD): Each build is kept ready for deployment
- Continuous Deployment: Every changes that passes the automated tests is deployed directly to production

• Comparison of Standard Software Development and Agile

Standard Methods follow a rigid, sequential approach, like the Waterfall Model, where each phase must be completed before moving to the next. They are best suited for large, stable projects with well-defined and unchanging requirements.

Instead, Agile Methods takes an iterative and incremental approach, emphasizing flexibility, collaboration and continuous delivery. Development happens in short cycles, allowing for adaptable requirements based on feedback. Agile Methods are ideal for dynamic projects with changing requirements