SHUBHANGI GUPTA
2K15-SE-071

SPM ASSIGNMENT-1

RATIONAL UNIFIED PROCESS

The fundamental purpose of rational unified process is to provide a model for effectively implementing commercially proven approaches to development, for use throughout the entire software development life cycle. Taking elements from other iterative software development models, Rational Unified Process framework was initially created by the Rational software corporation, which was brought out by IBM in 2003.

The Rational Unified Process is not a concrete developmental model but rather is intended to be adaptative and tailored to the specific needs of your project, team or organisation. RUP is based on a few fundamental ideas such as phases of development and the building blocks which define who, what, when and how development will take place.

Key Features of RUP :—

1. Iterative software development : The software is developed through a series of iterations. This allows the customer to provide continuous feedback after every iteration and hence reduces risks and increases quality of software. Iterative development ensures & helps in monitoring the schedule and budget of the project and also makes it easier to accomodate changing requirements.

2. Effective requirement elicitation : RUP promotes the use of an elicitation technique popularly known as use case approach. Every requirement is verifiable and traceable and every piece of design may be traced back to one or more requirements in the software. It ensures that effective testing is carrying by generating test cases from use cases. These test cases are used in 'acceptance testing' and ensure that the final software produced fulfills the user requirements.

3. Visual modelling : It creates models to solve problems around the real world. It provides an abstraction by hiding non-essential details and building models that portray different views of the system. Models are an efficient way to understand, visualize and document the real world objects. Modelling safeguards the system by providing understandable and manageable requirements, better software designs and flexible software.

4. Use and development of Reusable components: RUP supports component based software engineering. A component is an independant subsystem that fulfills a goal in a clear fashion. The wellcoded and thoroughly tested existing components are used and new well designed components are ~~used and~~ created to promote reuse.

5. Ensuring quality of a system: The cost to fix defects increases drastically when one moves from the requirements phase to the testing phase and finally to the maintainance phase. Hence quality must be continuously assessed in order to produce a flexible and manageable system. RUP provides guidelines for continuous quality assessments, product evaluations and process monitoring. It puts emphasis on building quality into the system from its inception

6. Change control and management : RUP provides guidelines for managing, controlling and tracking changes in order to enable effective iterative software development.

7. Automated Testing : Automative Testing does repetitive testing, unattended without any human intervention. Both functional and non-functional requirements can be tested by using a tool. This saves much time, effort and resources.

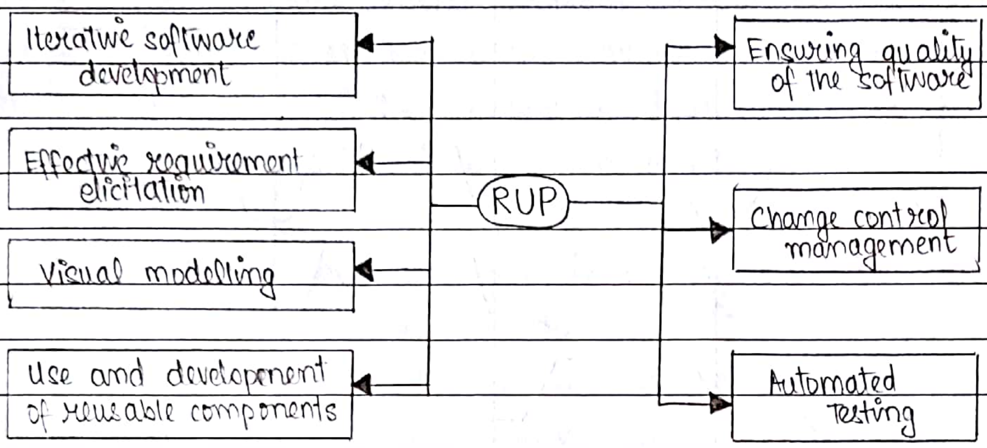

Iterative software development
Effective requirement elicitation
Visual modelling
Use and development of reusable components
RUP
Ensuring quality of the software
Change control management
Automated Testing

FIGURE 1: KEY FEATURES OF RUP.

RUP Structure:—

The RUP process is divided into 2 structures:

1. STATIC STRUCTURE – It provides the process description in terms of roles, activities, artifacts, disciplines and workflows.

2. DYNAMIC STRUCTURE – The dynamic aspect of the process can be viewed in terms of iterative development.

The structure of RUP is shown in figure 2. The vertical dimension depicts the static structure of the process and the time dimension is carried out as it depicts the dynamic structure of the process. Each activity on the time dimension is carried out iteratively. As seen in the figure implementation starts much later in the S/W development. Most of the requirements are pre-defined in the early phases, but some new requirements may be added in the later stages of development. Testing is carried out throughout the S/W life cycle, whereas ~~development~~ deployment activities begin much later in the S/W development.

Although the focus on activities keeps on increasing and decreasing throughout the S/W lifecycle, they can be carried out at any time during the S/W life cycle

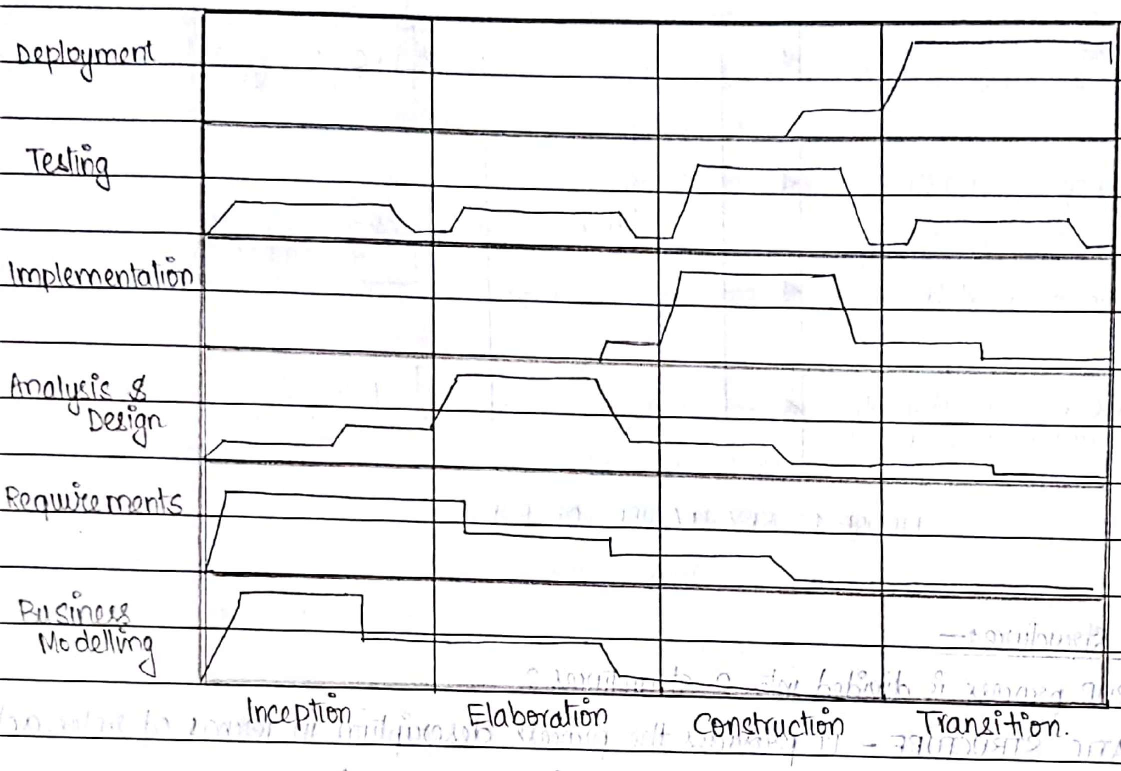
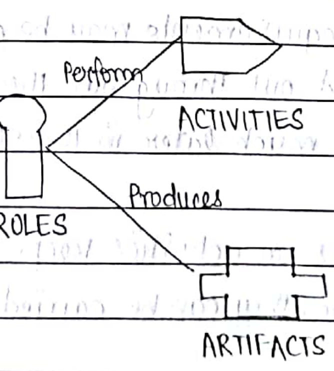


FIGURE 2: STRUCTURE OF AN RUP

STATIC STRUCTURE OF RUP :—

The static structure of RUP describes what should be produced, who is incharge of producing it, how production takes place and when production is complete. Five major elements form the static structure of the RUP :

Roles (WHO)
Activities (HOW)
Artifacts (WHAT)
Disciplines
Workflows (WHEN)



1. Roles : A role describes functions or position that a person is expected to have in a project. The functions are known as activities that a person has to perform in order to achieve an artifact. A person may have many roles. Roles of a person may keep on changing depending upon the project.

2. Activities : Activities are the work performed by a person in a specific role to produce the required result from the system. It is expressed in terms of design or creation of an artifact such as use cases, object, state chart, etc... An activity may be further decomposed into various subactivities.

3. Artifacts : Artifacts are the outputs produced during the development life cycle of the software. They may be end products produced or inputs used by the activities while developing the S/W. An activity is performed to produce an artifact which may include SRS, Use Case model, Test Plan, Source Code, ...

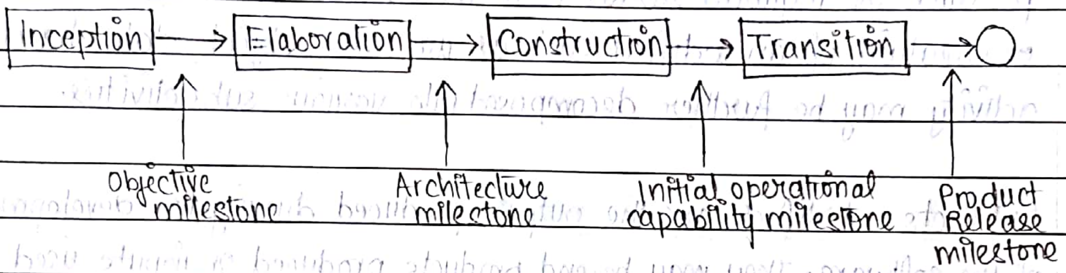
4. Disciplines : Disciplines are used to organize a set of activities. RUP consists of 6 major disciplines:
   - Business Modelling
   - Requirements
   - Analysis & Design
   - Implementation
   - Testing
   - Deployment

   These are traced again and again throughout the S/W Development Life Cycle.

5. Workflows : These represent a diagrammed sequence of activities in order to produce observable value and artifacts.

DYNAMIC STRUCTURE OF RUP :-

The dynamic structure deals with the lifecycle or time dimension of a project. It provides a structured approach to iterative development dividing the project into 4 phases : Inception, Elaboration, Construction and Transition. Each phase contains one or more iterations which focus on producing the technical deliveriables neccessary to achieve objectives of that phase.



1. Inception Phase: Establish a good understanding of what system to build by getting a high level understanding of all the requirements and establishing the scope of the system. Mitigate many of the business risks, produce the business case for building the system and get ~~by~~ buy-in from all stakeholders on whether to proceed with the project.

2. Elaboration Phase: Takes care of many of the most technically difficult tasks such as design, implement, test and baseline an executable architecture including subsystems, their interfaces, key components and architectural mechanisms such as how to deal with inter-process communication and persistency. Addresses major technical risks such as resource contention risks, performance risks, data security risks ~~and~~ by implementing and validating actual code.

3. Construction Phase: Do most of the implementation as you move from an executable architecture to the first operational version of the system. Deploy several internal and alpha releases to ensure that the system is usable and addresses user needs. End the phase by deploying a full functional beta version of the system including installation and supporting documentation and training material.

4. Transition Phase: Ensure that s/w addresses the needs of its users. This includes testing the product in preparation for release and making minor adjustments based on user feedback. At this point in lifecycle, user feedback focuses mainly on fine-tuning the product, configuration, installation and usability issues.

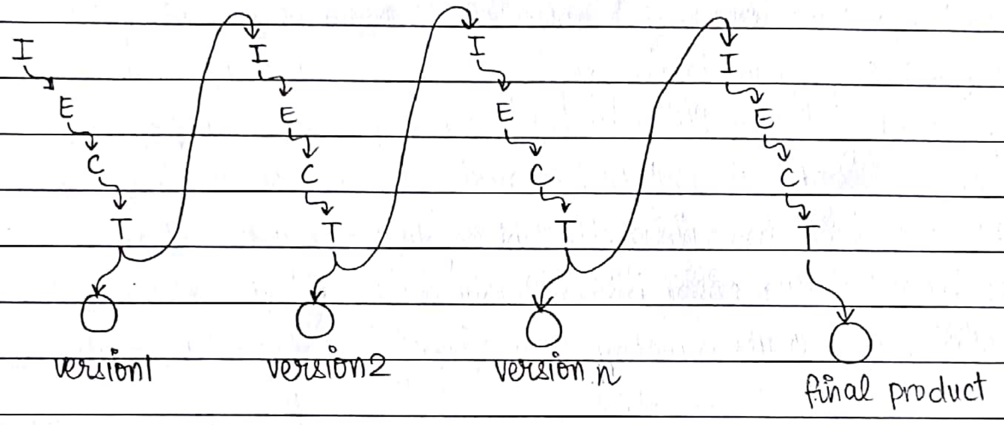


Figure 3: Iterative RUP

ADVANTAGES OF RUP:—

- Allows for the adaptive capability to deal with changing requirements throughout the development life cycle.
- Emphasizes the need and proper implementation of accurate documentation
- Diffuses potential integration headaches by forcing integration to occur throughout the development, specifically within the construction phase where all other coding takes place.

DISADVANTAGES OF RUP :—

- Heavily relies on proficient and expert team members since assignment of activities to individual workers should produce tangible, pre-planned results.
- Given the emphasis of integration throughout the developmental process, this can also be detrimental during testing or other phases, where integrations are conflicting
- RUP is a fairly complicated process/model. Often proper implementation of RUP can be challenging for many organisations, particularly for smaller teams or projects.