Requirements Engineering, Scrum and EDAP

**Goal oriented Requirement Engineering**

“A goal is an objective the system under consideration should achieve”

“A goal under responsibility of a single agent in the software-to-be becomes a requirement whereas a goal under responsibility of a single agent in the environment of the software-to-be becomes an assumption”

In Goal oriented Requirement Engineering, the desired properties of a System are described with the help of goals: “A goal is an objective the system under consideration should achieve”. Goals can be detected by asking the questions why, how and how else. To define the requirements for the system, goals are refined and broken down into subgoals. A requirement is then defined as a “goal under responsibility of a single agent in the software-to-be”. An Agent is an “active component[s] of the system, such as human, devices, and software”. Not only the system itself can be described with goals and requirement, the environment of the system also has goals, “a goal under responsibility of a single agent in the environment of the software-to-be becomes an assumption”. Refining goals also includes the resolution of conflicts between goals, requirements or assumptions. “Goals have been recognized to provide the roots for detecting conflicts among requirements and for resolving them eventually”

**Detailed:**

**Requirement Engineering**

Requirement Engineering is concerned with the definition of requirements to a system that will be developed. The goal is to arrive at a set of requirements that completely describe the functionality, the constraints, what the system should and should not do. The requirements guide the implementation and serve as rationale for the systems properties.

There are three activities in Requirements Engineering: Requirements Elicitation, Requirements Analysis, and Requirements Management. In Requirements elicitation, requirements are collected. There are many sources for requirements to be consulted/investigated. A big portion will come from the client. If for example the client wants the system to provide a user interface, one requirement will be: The system shall provide a user interface. Other sources are for example: industry best practices, available resources or security standards. The requirements can be quite specific, or broader. Some requirements might stand in conflict with each other.

In the next step, requirements analysis, the requirements are refined, categorised, broken down and conflicts are solved. Requirements can be categorised into two groups: functional and non-functional requirements. Functional requirements are concerned with the functions a system should have. “The System shall provide a login function” is a functional requirement. Non-functional requirements describe the qualities a system should have. This could be accuracy, performance or security. “The system shall be modifiable to include new functionality in the future” is such a requirement. Requirements that contain multiple requirements can be refined by breaking them down into smaller ones. A requirement posed by the client could be “The system shall have a secure login mechanism”. This can be broken down into: “The system shall have a login mechanism” and “The login mechanism shall be secure”. A broad requirement can lead to more specific requirements. When multiple requirements stand in conflict, that conflict has to be resolved. For example, a system should accommodate a number of users, but it should also run on resources that cannot support that number of users. To solve this problem, the maximum number of users could be reduced.

In Requirement Management, requirement change is managed. Requirements can change, sometimes the client changes them, sometimes the environment changes. These changes have to be included, so that the requirements always match up with the finished product. To determine whether to System fulfils its requirements, it has to be tested and verified.

**Goal oriented Requirement Engineering**

Goal oriented requirements engineering is an approach to requirements engineering that uses goals to elicit and analyse requirements. “A goal is an objective the system under consideration should achieve”. Goals can be detected by asking the questions why, how and how else. These questions lead to different levels of goals. Asking why leads to the highest-level questions, “it helps to discover the objectives and rationale behind the goals”. The how question leads to lower level goals of technical nature. Asking the how else question “helps to identify the alternates to satisfy higher level goals”. These questions are part of the requirement elicitation phase.

To break down and refine goals, the concept of agents is used. Agents are “active components of the system, such as humans, devices, and software”. A goal is broken down, until only one agent is responsible for it. The goal then becomes a requirement. Not only the system itself can be described with goals and requirement, the environment of the system also has goals, “a goal under responsibility of a single agent in the environment of the software-to-be becomes an assumption”

**Scrum**

In Scrum, software is developed incrementally. The requirements and tasks are listed in a Product Backlog. Each increment, called Sprint, a number of items in the Product Backlog are worked on. Before each sprint, there is the Sprint Planning meeting. The items from the backlog are selected and their implementation planned. Each day during the sprint, daily scrum meetings are held where progress and impediments can be discussed and the work for the day is planned. After the sprint, the increment is reviewed in the Sprint Review and improvements are discussed in the Sprint retrospective.

**EDAP**

In EDAP, the planned system, its actors/stakeholders and its environment are first described, to map out values, relationships and interests. Then, biases are listed, which describe given circumstances that can impact the decisions that will be made about the system. In the next phase, conflicts between values, goals and interests are explored and analysed. Pros and Cons for courses of action are specified. From this information, it is decided whether to stop and find a different solution or to go forward. In that case, the technical feasibility is analysed. The final step is testing if the finished System holds the standards that were defined.

**Combining the three/ common features**

In all three, there is some form of goals. In requirement engineering its goals and requirements, in scrum, there are backlog items and in edap, there are ethical values and other goals. These goals dictate what should be done, sometimes they can contradict each other, which causes a conflict. For example, the goal: the system should provide a very personalised user experience can come into conflict with the value: the users data should be private. In each strategy, these goals are refined and subgoals are specified. In scrum, backlog items can be divided into tasks. In edap, a broad ethical value can be made more specific to the system at hand. Both goal oriented requirement engineering and edap are strategies to define requirements to the system. Scrum is a strategy to take such requirements and guide their implementation. Requirements engineering and edap should therefore be integrated into the scrum process whenever requirements or goals are added to the backlog.