**POM 02 – Project Organization**

**Team Formation**

*Stages of Team Development:*

* Stage 1: **Forming** (goal alignment, get to know each other, high dependence on leader for guidance and direction)
* Stage 2: **Storming** (member forms opinions of each other, ideas compete for consideration)
* Stage 3: **Norming** (resolving conflict, roles are accepted, team becomes more independent from its leader)
* Stage 4: **Performing** (empowering behaviours, most effective stage, trust, team leader less important in decision making and problem resolution)

*Difference between group and team:*

* **Group**: number of people that have some relationship to one another
  + Loosely connected
  + Not focus on specific outcomes
  + Every individual works on his own
* **Team**: Any group of people involved in the same activity with a common goal, especially referring to sports and work
  + Strongly connected
  + Focus on specific outcome
  + Members need to work together

**Project Management Terminology**

*Project Deﬁnition:*

A project is an undertaking, limited in time, with a clear goal and a speciﬁc budget, requiring a concerted effort

A **project** consists of:

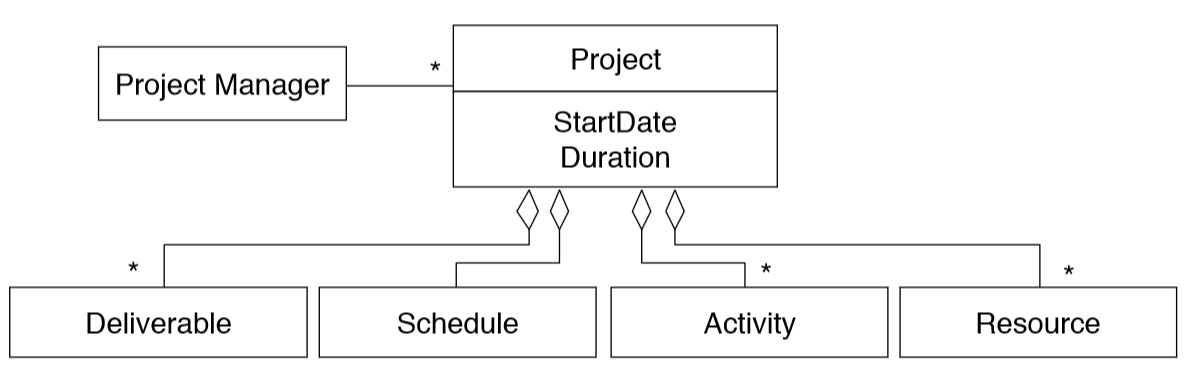
* Start date and duration
* Set of deliverables to a client
* Schedule
* Technical and managerial activities required
* Resources consumed by the activities

A project is managed by a **project manager** who:

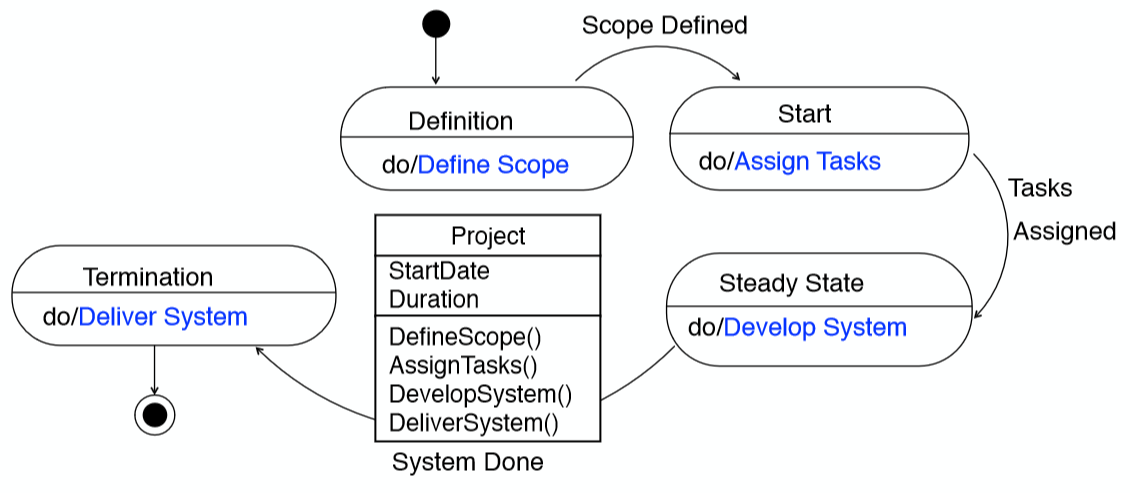
* Administers the resources
* Maintains accountability
* Makes sure the project goals are met

**Post mortem analysis and retrospectives help to improve yourself in future projects**

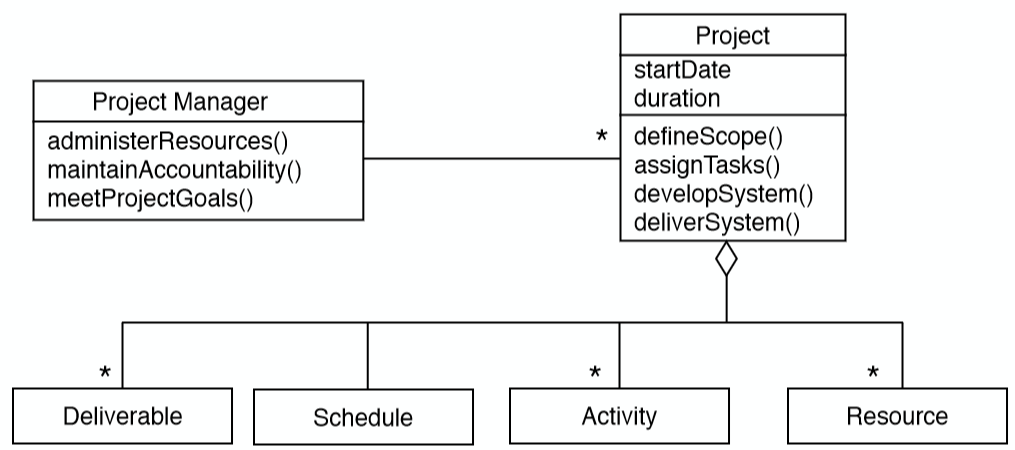
*Initial Object Model:*



*Dynamic Model:*



*Reﬁned Object Model:*

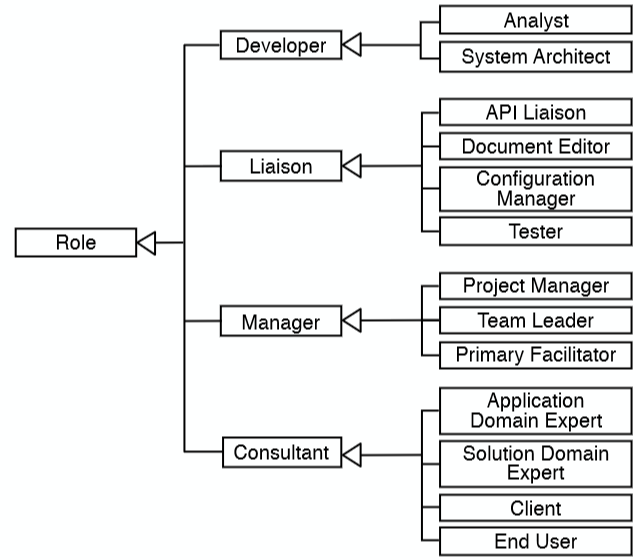


*Role:*

A role deﬁnes a set of responsibilities, a responsibility is a duty or task a person is required to do. Examples of roles and corresponding responsibilities:

* **Project manager** (administer the resources, make sure the project goals are met)
* **Analyst** (analyse the application domain, create a taxonomy of the domain abstraction)
* **System architect** (decompose the system into subsystems, choose a software architectural style
* **Tester** (design and implement tests)

*A Role Taxonomy: Types of Roles:*



*Roles and Responsibilities:*

* Responsibilities are assigned to roles
* Roles are assigned to people
* People are assigned to teams

*Assignments of Roles to Participants:*

* **One-to-One** (ideal but rare)
* **Many-to-Few** (each project member assumes several “hats”, danger of over-commitment)
* **Many-to-“Too-Many”** (some people don’t have significant roles, lack of accountability, loosing touch with project)

*Problems:*

* **Incompetence** (wrong person fills the wrong role)
  + **Peter Principle**
  + “Employees who perform their roles in a hierarchy with competence are promoted to a higher level until they reach a level where they are no longer competent. There they remain forever”
* **Useless roles** (role exists only to minimize damage control)
  + **Dilbert’s Law**
  + “Companies tend to systematically promote their least-competent employee to management (generally middle management), in order to limit the amount of damage they are capable of”
* **Increase of Bureaucracy** (role swells unnecessarily simply because it can)
  + **Parkinson’s Law**
  + „Work expands to ﬁll the time available for its completion“ 2 reasons:
    - 1) “Ofﬁcials want to multiply subordinates, not rivals”
    - 2) ”Ofﬁcials make work for each other”

*Refactored Solution Heuristics:*

* **Dealing with incompetence:**
  + It makes little sense to take your most brilliant engineer and have him or her manage people and budgets
* **Dealing with useless roles:**
  + Put individuals to work in their core competencies
* **Dealing with increased bureaucracy:**
  + Improve estimation
  + Don’t wait until the last minute

*Key concepts for mapping roles to people:*

* **Authority**: The ability to make binding decisions between people and roles
* **Responsibility**: The commitment of a role to achieve speciﬁc results
* **Accountability**: Tracking a task performance to a speciﬁc person
* **Delegation**: Binding a responsibility assigned to one person (including yourself) to another person

*Delegation:*

Three reasons for delegation:

1. **Time Management**: To free yourself up for other tasks
2. **Expertise**: The most qualiﬁed person makes the decision
3. **Training**: To develop another person’s ability to handle additional assignments

* **You can delegate authority, but you cannot delegate responsibility**
* **You can only share responsibility**

*Task:*

A task describes the smallest amount of work monitored (tracked) by the project manager

* Typically less than 2-4 working days effort

*Task Sizes:*

* You may not know how to decompose the problem into tasks at ﬁrst
* Finding the appropriate size has to be learned
  + To-do lists from previous projects
  + If a task size is too large, the task should be renamed into an activity
  + An activity is decomposed into smaller tasks that allow monitoring

*Activities:*

* An activity is a major unit of work
* Culminates in a project milestone
* Activities can also have internal checkpoints
* Activities allow to separate concerns
* Precedence relations often exist among activities

**Examples:**

Planning, Requirement Elicitation, Analysis, System Design, Configuration Mgmt., …

*Unit of Work:*

* Activities are often grouped again into **higher-level** activities
* **Unit of Work:** A task or an activity that contains other tasks and lower-level activities

*Project Function:*

An activity that spans the entire duration of a software project.

**Examples:**

* project management, documentation, conﬁguration management, testing, continuous integration, release management;

*Work Package:*

* A **task** or **activity** is speciﬁed by a **work package** which contains:
  + Description of work to be done
  + Preconditions for starting, duration, required resources
  + Work products to be produced, acceptance criteria for it
  + Risks involved
* A work package must have completion criteria
  + Includes the acceptance criteria for the work products produced by the task or activity

*Work Product:*

A **work product** is the visible outcome of a unit of work (model, presentation, piece of code)

Work products that have to be given to the customer are called **deliverables**

