The Unified Process (UP)

To effectively and successfully apply the UP, we must understand collaborations, contexts, and interactions. As an effort or project leverages the UP, collaborations focus on the elements of the project, context focuses on the process framework for the project, and interactions focus on the execution of the project. Figure 1 shows the various elements of the UP.

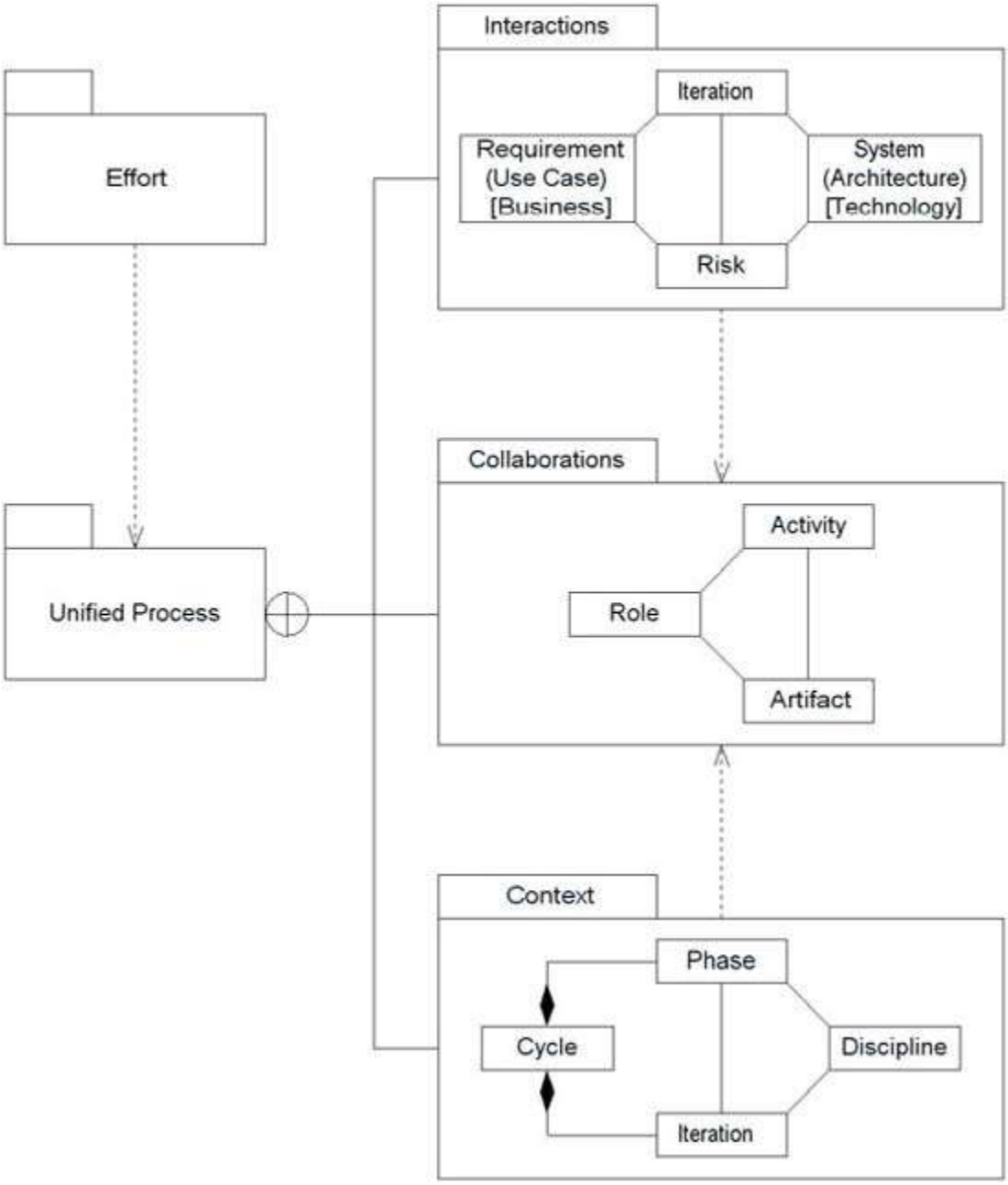

Figure 1: Elements of the Unified Process (UP).

Collaborations

A collaboration involves an interaction within a context. A collaboration captures who does what activities (how) on what work products. Thus, it establishes the elements of a project.

A role is an individual or team who has responsibility for activities and artifacts. An activity is a unit of work, composed of steps, that is performed by a role. An artifact is an element of information that is the responsibility of a role and that is produced or consumed by activities. The UP defines numerous roles, artifacts, and activities.

Contexts

A context emphasizes the structural or static aspect of a collaboration, the elements that collaborate and their conglomeration or spatial relationships. A context captures when and where such activities should be done and work products produced and consumed. Thus, it establishes the context for a project. Figure 2 shows the context established by the UP.

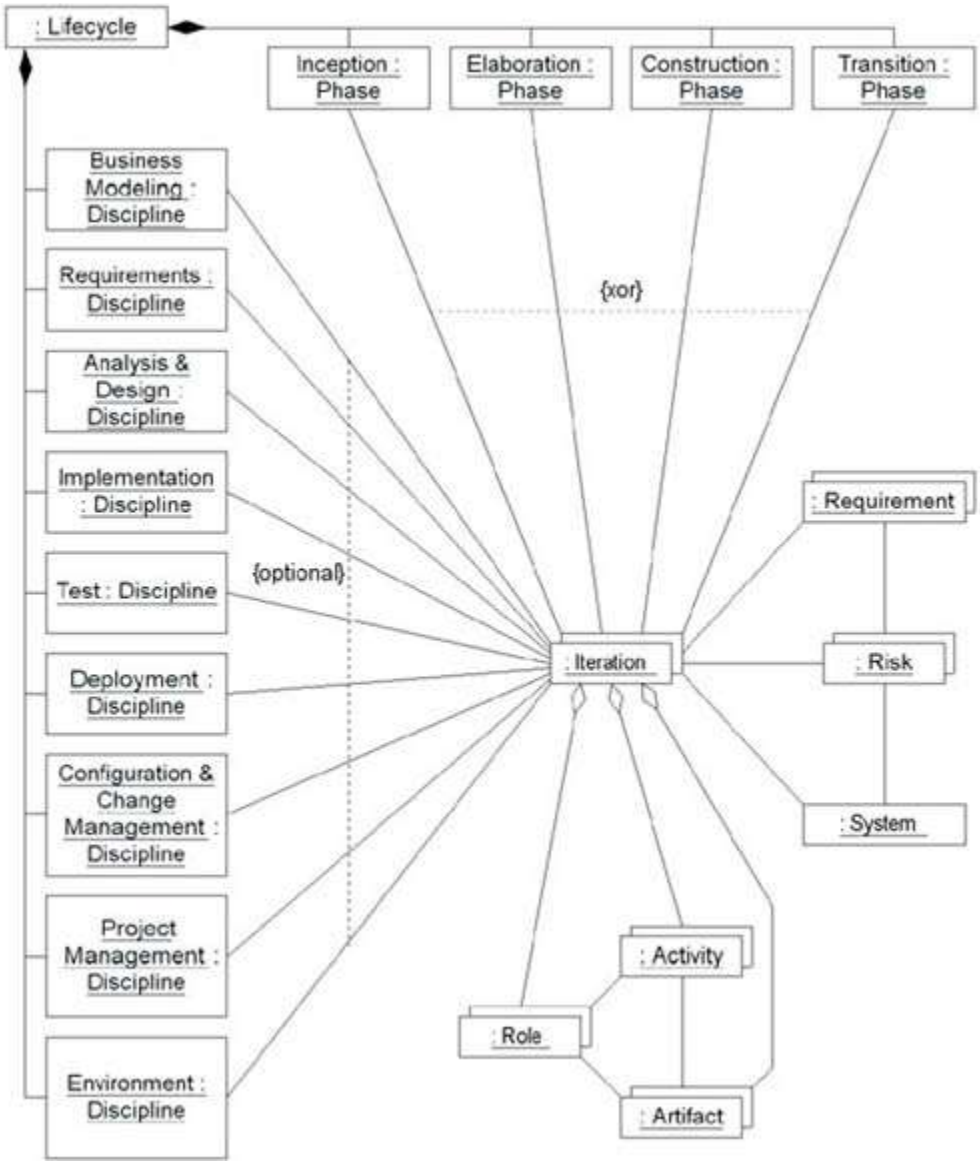

Figure 2: Context established by the Unified Process (UP).

A project requires a management perspective to manage the effort and a technical perspective to execute and perform the technical work. The lifecycle of a project is composed of phases wherein iterations involve disciplines. A development cycle is composed of sequential phases resulting in a major system release called

a system generation. For example, system generations may include versions 1.0, 2.0, 3.0, and so forth. A phase is a major milestone, a management decision point focused on managing business risk. Phases embody the macro-level problem-solving process. An iteration is a minor milestone, a technical decision point focused on managing technical risk, resulting in a minor system release called a system increment. For example, system increments may include versions 1.1, 1.2, 2.5, and so forth. Iterations embody micro-level applications of the scientific method. A discipline is an area of concern or theme wherein workflows describe the flow of work and wherein workflow details describe the collection of activities (with their associated roles and artifacts) often done together.

The UP defines the following four phases:

- The Inception phase, concluding with the Objective milestone, focuses on establishing the project's scope and vision; that is, establishing the business feasibility of the effort and stabilizing the objectives of the project.
- The Elaboration phase, concluding with the Architecture milestone, focuses on establishing the system's requirements and architecture; that is, establishing the technical feasibility of the effort and stabilizing the architecture of the system.
- The Construction phase, concluding with the Initial Operational Capability milestone, focuses on completing construction or building of the system.
- The Transition phase, concluding with the Product Release milestone, focuses on completing transitioning or deployment of the system to the user community.

The UP defines the following three supporting disciplines:

- The Configuration & Change Management discipline focuses on managing the configuration of the system and change requests.
- The Project Management discipline focuses on managing the project.
- The Environment discipline focuses on the environment for the project, including the process and tools.
- The UP defines the following six core disciplines:
- The Business Modeling discipline focuses on understanding he business being automated by the system and capturing such knowledge in a Business model.
- The Requirements discipline focuses on understanding the requirements of the system that automates the business and capturing such knowledge in a Use-case model.
- The Analysis & Design discipline focuses on analyzing the requirements and designing the system and capturing such knowledge in an Analysis/Design model.
- The Implementation discipline focuses on implementing the system based on the Implementation model.
- The Test discipline focuses on testing (evaluating) the system against the requirements based on the Test model.
- The Deployment discipline focuses on deploying the system based on the Deployment model.

The distribution of effort across phases, iterations, and disciplines focuses on addressing business and technical risks. During the Inception phase, most of the effort is distributed across the Business Modeling and Requirements disciplines. During the Elaboration phase, most of the effort is distributed across the Requirements, Analysis & Design, and Implementation disciplines. During the Construction phase, most of the effort is distributed across the Analysis & Design, Implementation, and Test disciplines. During the Transition phase, most of the effort is distributed across the Test and Deployment disciplines. The supporting disciplines are generally distributed throughout the four phases. The overall objective is to produce the resulting system; therefore, all of the core disciplines are engaged as soon as possible without introducing risk to the project; that is, practitioners are responsible for determining which disciplines to engage and when they should be engaged.

Interactions

An interaction emphasizes the behavioral or dynamic aspect of a collaboration, the elements that collaborate and their cooperation or temporal communication. An interaction captures when and why such activities should be done and work products produced and consumed. Thus, it establishes the execution of a project as it is governed by various forces.

As minor milestones occur within major milestones, technical decision points occur within management decision points such as to align technical tactics and operations with business strategy and objectives -- essentially, establishing a bridge between business and technical forces.

An iteration is a step or leg along a path or route to a destination. An iteration is planned and is not ad hoc, has evaluation criteria, and results in demonstrable progress. An iteration is iterative in that it is repetitive and involves work and rework, incremental in that it is additive and involves more than rework alone, and parallel in that work may be concurrent within the iteration.

A use-case is a functional requirement. For example, functionality to login or logout of a system, input data, process the data, generate reports, and so forth. As the UP is use-case driven, use cases drive or feed iterations. That is, iterations are planned and evaluated against "chunks" of functionality (or parts thereof) such as to manage agreement with users and trace project activities and artifacts back to requirements. Thus, accounting for business forces by planning and evaluating iterations against functional requirements. Non-functional requirements (usability, reliability, performance, and other such characteristics) are incrementally considered as use cases evolve through the disciplines.

A system has an architecture. For example, the architecture of a system includes a collection of elements and how they collaborate and interact, including various subsystems for handling security, input and output, data storage, external communications, reporting, and so forth. As the UP is architecture-centric, iterations focus on architecture and evolving the system. That is, iterations demonstrate progress by evolving a puzzle of "chunks" such as to manage the complexity and integrity of the system. Thus, accounting for technical forces by demonstrating progress via the production and evolution of the real system.

A risk is an obstacle to success, including human, business, and technical concerns or issues. For example, human risks include having insufficient, untrained, or inexperienced human resources, and so forth; business risks include having insufficient funding, time, or commitment from the business community, and so forth; and technical risks include having an insufficient understanding of the requirements or technology, using unproven technology, using technology that will not sufficiently address the requirements, and so forth. As the UP is risk-confronting, iterations confront risk and leverage feedback from previous iterations to confirm progress and discover other unknown risks. That is, iterations confront risk that is derived from use cases and architecture such as to achieve project success,thus reconciling business and technical forces.

An iteration is a time-box with a fixed beginning and end wherein a collection of collaborations are planned, executed, and assessed in order to progressively demonstrate progress. The beginning and end are negotiated among stakeholders, management and technical members of the project community who impact and are impacted by the effort. Use cases that feed an iteration are selected based on the highest risks they confront. A use case may evolve across any number of iterations and may evolve through any number of core disciplines in an iteration. An iteration results in one or more intermediate builds or operational versions of the system. An iteration results in a single internal or external baselined and evaluated release of the system. The feedback and lessons-learned gained from an iteration feed into future iterations. Within an iterative approach, metrics and estimates are also iteratively derived, and trends across iterations form the basis for metrics and estimation for the overall effort. The duration of an iteration is inversely proportional to the level of risk associated with the effort. As iterations execute, they only minimally overlap. Development cycles and phases may also be time-boxed; as development cycles, phases, and iterations are planned, the further the plans are in the future, the less accurate the estimates.

Although iterations are composed of the same disciplines as a "pure waterfall" approach, there are key distinctions. A waterfall approach aims for one hundred percent completeness of activities and artifacts of a discipline before proceeding to the next discipline; however, an iterative approach involves iterative collaboration and aims for incremental refinement and evolving levels of detail of artifacts throughout the lifecycle. A waterfall approach does not offer explicit opportunities for partial deployment of a system or explicit opportunities for introducing change into the lifecycle, and is therefore quite reactive to change; however, an iterative approach does offer explicit opportunities for partial deployment of a system at the end of an iteration and explicit opportunities for introducing change into the lifecycle at the end of an iteration and before the next iteration, and is therefore quite proactive or responsive to change. A waterfall approach progresses serially through disciplines; however, an iterative approach may progress forward or backward across phases to change focus and involves various disciplines in order to address risk.

Iterations

To effectively and successfully apply the UP, we must understand iterations and how they are applied in linear, sequential, and iterative approaches.

An iteration is planned, executed, and evaluated. Use cases and risks are prioritized, and use cases are ranked against the risks they mitigate. When planning an iteration, those use cases that address the highest risks and can be accommodated given the iteration's limiting factors (funding, time, resources, and so forth) are selected for driving the iteration. When executing an iteration, use cases evolve through the core disciplines and the system and its architecture evolve.

However, use cases need not evolve through every core discipline in a single iteration. When evaluating an iteration, actual results are compared against the planned objectives of the iteration, and plans and risks are updated and adjusted. The overall objective is to produce the resulting system; therefore, all of the core disciplines are engaged as soon as possible without introducing risk to the project; that is, practitioners are responsible for determining which disciplines to engage and when they should be engaged.