# RUP® at a glance

**Proven. Practical. Flexible.**

**Best practices:**

- **Develop Iteratively**  
  Plan projects into iterations, a number of weeks. Each iteration should result in an executable part of the system. Major technical risks should be solved in early iterations. This will give you control of scope, budget, and schedule earlier in the project.

- **Manage Requirements**  
  Analyze and understand customer and user problems and needs. Elicit, document (Use Cases and Supplementary requirements) and manage the changing requirements.

- **Use Component Architectures**  
  Build executable architectures based on components to reduce system size and complexity, and make the systems more robust and resilient.

- **Model Visually (UML)**  
  Make visual models to improve communication about the system requirements and design and keep them consistent. UML (Unified Modeling Language) provides notation for modeling different perspectives on the system.

- **Continuously Verify Quality**  
  Plan and run testing and other verification activities from the beginning of the project to find deficiencies while it is still feasible to correct them, and allowing you to avoid similar mistakes.

- **Manage Change**  
  Manage integration and control versions and changes to code, documents and models to keep them consistent. Provide secure workspaces for team members.

Solving difficulties we all experience in software development:

- **Maintain a high level of efficiency**
- **Address the user/client business needs**
- **Focus on the most essential capabilities**
- **Keep the project on budget and schedule**

<table>
<thead>
<tr>
<th>Disciplines</th>
<th>Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Modeling</td>
<td>INCEPTION</td>
</tr>
<tr>
<td>Requirements</td>
<td>ELABORATION</td>
</tr>
<tr>
<td>Analysis &amp; Design</td>
<td>CONSTRUCTION</td>
</tr>
<tr>
<td>Implementation</td>
<td>TRANSITION</td>
</tr>
<tr>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>Deployment</td>
<td></td>
</tr>
<tr>
<td>Configuration &amp; Change Management</td>
<td></td>
</tr>
<tr>
<td>Project Management</td>
<td></td>
</tr>
<tr>
<td>Environment</td>
<td></td>
</tr>
</tbody>
</table>

In an iteration, you walk through all disciplines.

Every iteration results in an executable release!

Glossary:

- **Discipline**: A logical grouping of activities that make a common concept. A discipline contains roles, activities, artifacts, concepts and guidelines that are used throughout.

- **Inception**: Define scope of project, estimate cost, develop schedule, identify risks. Identify high-level requirements.

- **Elaboration**: Elaborate all major technical risks, build architecture, detailed requirements, plan project, cost and schedule.

- **Construction**: Build the system, optimize performance, develop first operational version of the system.

- **Transition**: Transition the system to end-users, maintenance and customization, finishing.

- **Role**: Define the behavior and responsibilities of an individual or a set of individuals working together as a team.

- **Artifact**: A piece of information that is produced, modified, or used by the activities in RUP. Documents, models, and code are examples of artifacts. The responsibility of one role.

- **Activity**: A piece of work that may be assigned to perform. Repeated as necessary, throughout the project. The responsibility of one role.

- **Controlled Iterative development**:  
  - An iteration is a loosely sequential set of activities from all disciplines.
  - A major iteration results in an executable release (software or design of the project), i.e., you develop a part of the final system.
  - High quality code should be developed in early iterations.
  - The schedule for an iteration should be regarded as fixed, and the content actively managed to meet that schedule.

- **Iteration**: A loosely sequential set of activities from all disciplines.

- **Major iteration results in an executable release (software or design of the project), i.e., you develop a part of the final system.

- **High quality code should be developed in early iterations.

- **The schedule for an iteration should be regarded as fixed, and the content actively managed to meet that schedule.**