

RTIS Software Development Approach

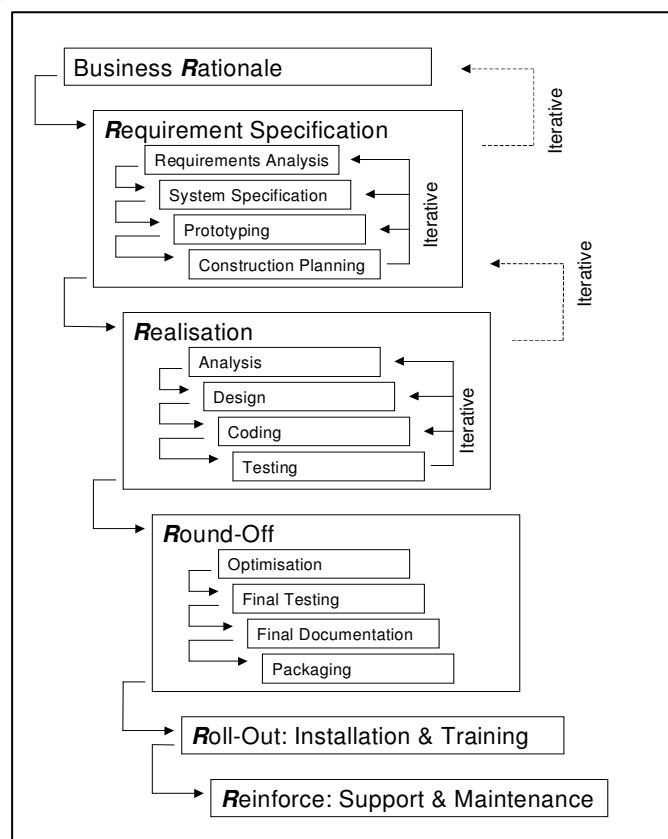
Project Success Factors and Use of Best Practices

RTIS offers clients a commitment to ensuring successful implementation of IT initiatives. We approach this by concentrating on factors that are crucial for project success, namely:

- Use of proven delivery processes, methods and tools
- Close Customer involvement
 - Understand the requirements and expectations of the client
 - Involve representative business and technology people
 - Obtain continuous client feedback throughout the project
 - Realise impact of technology solutions on people and their working practices
- Identify and deal with project risks
 - *Requirement risks* – Access to domain expertise is essential in establishing requirement specification and review of GUI prototypes. They need to be open-minded, have deep hands-on understanding and be readily available for questions.
 - *Technological risks* – Build prototypes early
 - *Skills risks* – Team selection, training and mentoring
 - *Political risks* – Identify ‘Roles’ - Sponsoring/buy-in established
 - *Temporal risks* – Project management in place, staged development

Recognised development best practices employed to leverage these success factors include:

- **Develop Software Iteratively** – Misunderstandings and inconsistencies are uncovered early, user feedback is encouraged and it helps project management
- **Requirements Management** – This is a systematic approach to finding, documenting, organizing, and tracking a system's requirements and changes to requirements. Keys to effective requirements management include maintaining a clear statement of the requirements and ensuring traceability to other requirements.
- **Use component architectures** – Components are non-trivial modules – groups of code or subsystems that fulfill a clear function. Architectures based around components tend to reduce the effective size and complexity of the solution, and so are more robust.
- **Model visually** – Visual modeling is the use of graphical design notations, such as the industry-standard Unified Modeling Language (UML), to capture software designs. Visual modelling aids the specification, analysis and design processes themselves, as well as providing documentation of the resulting design and analysis decisions that is concise and easily understood by stakeholders.
- **Continuously verify quality** – Poor performance and poor reliability are common results of an inadequate focus on software quality. An emphasis on quality is made throughout the project lifecycle, with testing conducted in each iteration.



Our generic software development process is given by the diagram, details may differ with the type of software product or system and the scale of the project.