

The Altimus Approach To Project Management

Introduction

A robust approach to project management is key to our ability to deliver high quality solutions to our customers on time and on budget. To facilitate this we use a project management methodology that has been tried, tested and developed through many projects over time.

The vast majority of our projects involve detailed interactions with our clients, and this is how it should be. However this does take some aspects of the project out of our control and thereby increases our risk. To mitigate this risk we are happy to extend our project management skills to cover the customer aspects of a project as well as our internal work. Should the customer choose this option the development part of the project would be delivered at a lower overall cost, as our risk is lower. There would however be an additional charge for the project management service we provide.

Altimus Project Management

With the Business Intelligence (BI) and Data Collaboration projects we are typically involved with it is often the case that the customer cannot write a detailed upfront specification of everything they want from a system. This can be for a number of reasons:

- Capability of the BI software not fully understood
- Source data is in a better or worse state than originally assumed
- Business changes during the project
- Interfaces to/from other systems change or are required
- Pressure from customers/suppliers to deliver something ‘special’
- Seeing the power of modern BI tools leads to the ‘that’s good, can we?’ response.

All these events point us towards a rapid application development model with frequent interaction with the customer for feedback and to guide future development along the correct path. At the same time we need to avoid a never-ending commitment to changes that result in cost and time overruns and overwhelming scope-creep.

Attaining a correct balance between the two is the real challenge for modern project management. This is what Altimus achieves with our project management methodology.

Methodology

The methodology fits into a 4 stage framework, known as PIRL:

Project start up

Identification

Realisation

Live running

Project Start Up

Actions in this relatively short stage will include

- Confirming the high level user requirements (as included in the quotation) or writing the high level user requirement
- Developing an outline project plan identifying the major phases of the project (should a phased implementation be required)
- Naming the key individuals for the project (Altimus and Customer)

The following stages would be repeated for a phased product.

Identification

In this phase we will identify the key deliverables from the user requirements and identify the following items:

- The data items required to achieve each deliverable
- The source for each data item
- How each data item will be obtained
- The likely data cleanliness of each data item
- Who is responsible for providing each data item
- The data consumer for each deliverable (be it person or system), and the owner of that deliverable

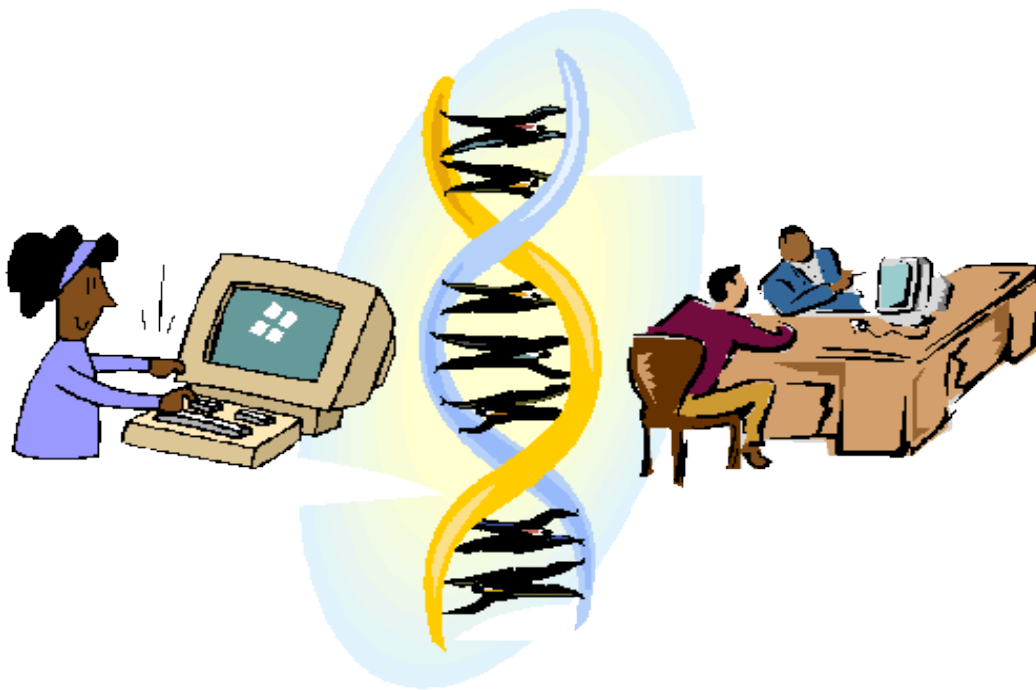
The outcome for this phase will be:

- A functional definition of each key deliverable
- For each interface (in or out) a detailed specification (be it data schema or UI design)

- A mapping document for the source of each data element required
- A gap analysis of where functionality will be impaired due to missing or dirty data
- A data cleansing process design to clean dirty data, if possible.
- A project plan identifying who (Altimus, Customer) is responsible for each part and crucially the amount of time allowed for the initial realisation and subsequent amendment.

Realisation

This covers the coding, testing and feedback of the development cycle and will inevitably be the longest part of the project. This phase will be based on the helical model of development [i]. This characterizes the rapid development in terms of a rising helix of interaction between developers and consumers:



In this model there are many touches between the developers on the left and the customer on the right. Each step represents iteration in the design and development process. By having many such touches the developers and customers stay close – bound together by the helix – and a rapid and mutually beneficial solution to issues is found.

The identification phase generated a functional definition of each deliverable. This will be refined through the helical process until the completed application meets the customer's needs – as they are perceived having **realised the benefits** of the Altimus solution.

This process is time-bound rather than content-bound (unless agreed otherwise) to ensure that the project meets its time and budget constraints. Clearly there may be give and take between the various deliverables, but the overall project must be contained.

The major risk of overruns lies in this phase. This is the reason that if Altimus provide project management services we are able to better manage the risk and thus provide lower cost development work and guarantees of cost and time budgets. Although obvious it needs to be stated that where an overall project plan requires input from the customer against a specific schedule this needs to be delivered to that schedule to ensure project deliverables are met.

The key outcome from this project step is a working solution ready for implementation.

The final step here is (wherever possible) a 'dress rehearsal'. This would take place approximately 1 – 2 weeks before live and include a test of all steps required to go-live. The outcome of this would be an agreement from the customer that the system is ready for go-live or a specific list of issues to be resolved. At this stage this must not include new functionality.

Live Running

This phase of the project covers the go-live process and will include some or all of the following as appropriate:

- Data conversion
- User training
- Software implementation to live system(s)
- User documentation
- System/Operational documentation
- Post go-live support

Project Documentation

The project documentation set will include some or all of the following. The detail and complexity of each will depend on the project size and who holds the project management role. Those highlighted in bold are Altimus' responsibility only when Altimus have ownership of the project management role.:

- **Statement of requirement.** A high level document containing the overall expectations of the project
- **Project Plan.** Either in MS Project or Excel depending on complexity. Who is responsible for what by when. This will be a living document, varying as the needs of the project but vary. It will be used as a baseline.

- **Functional specification.** Another living document, which reflects all the changes, decisions and requirements identified throughout the project. At the end of the project this will form the basis of system documentation.
- **Interface design document.** Detailed documentation of the interfaces (human and software) required by the project
- **Database design document.** A description of the databases, tables, fields, views and other database objects implemented for the project.
- **Action register.** This is a key document that will list the detailed tasks, issues, questions that need to be resolved. It will include who will resolve them and by when. When a decision is agreed the agreement will be recorded in the functional specification and referenced back to the action register.
- User and system documentation as required.

Timescales

Although there will clearly be a great deal of variation between the phases depending on the scale and complexity of an individual project typically we can say that the times would breakdown as:

- Project set-up : 10%
- Identification : 20%
- Realisation : 55%
- Live running : 15%

[i] Edmonds, Robin. The Helical Model of Software Development: DTI Conference on Software Methods: 1992
