



Promoting IT Efficiency

QAIassist
Blog Article
“V” Model

The "V" Model

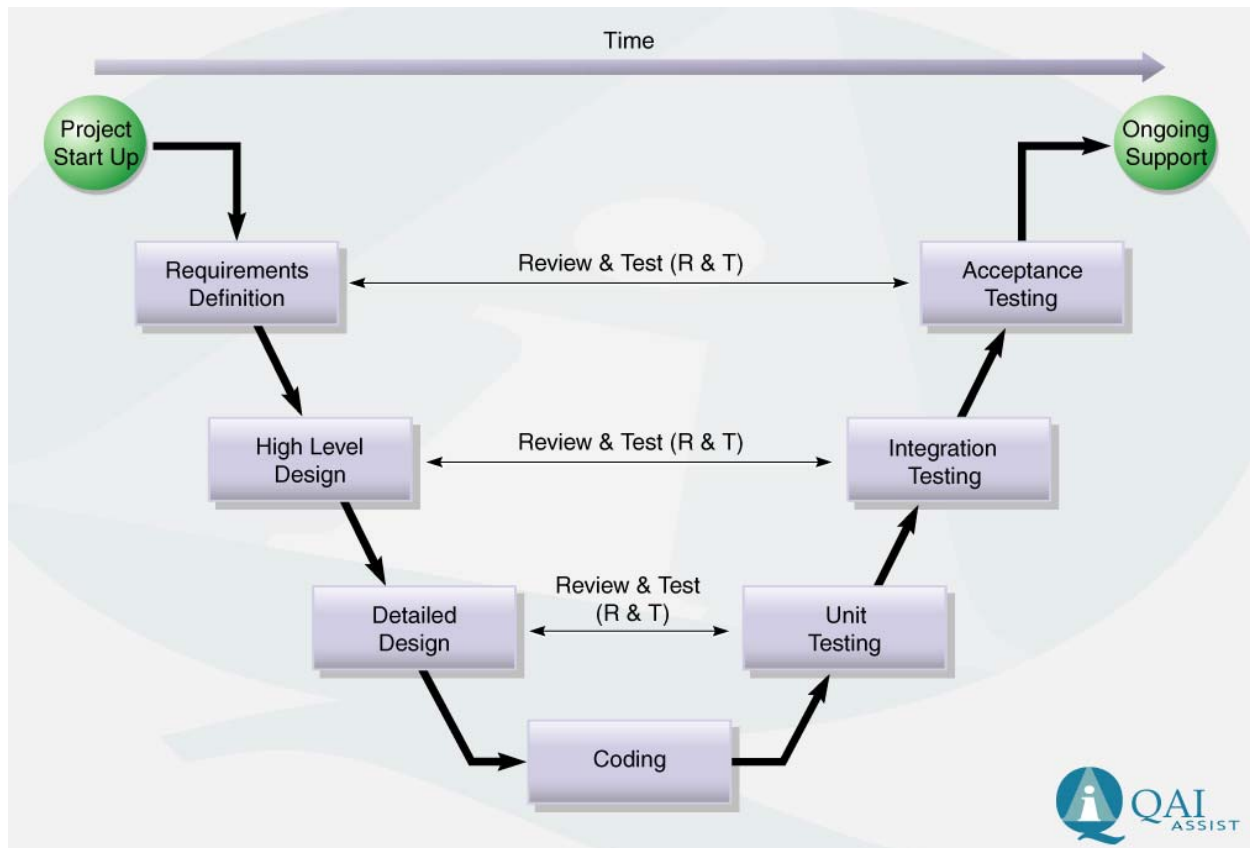
Context

From its inception Information Technology (IT) has recognized the significance and importance of developing and applying a set of "standards", "methodologies", "lifecycles" and "best practices" that can be leveraged by all practitioners. As the industry has evolved, the technologies have become more complex, increasingly faster, and forever changing, however, there remains a set of basic principles and concepts that are as applicable today as when IT was in its infancy.

One of initial concepts created and applied by early IT practitioners was the "V" Model. It was created to ensure project teams had a mechanism with which they could

- accurately define and refine user requirements
- design and build an application according to the authorized user requirements
- validate that the application they had built adhered to the authorized business requirements

The "V" model evolved in the 1960's – since that time various institutions and authors have revised, enhanced and extrapolated on it. It is possible to see a multitude of versions of the "V" model – each with its own customized terminology, phase names, and depictions. Though the IT industry has made significant strides since its inception, the principles defined by the "V" model are as applicable today as when the model was first created.



"V" Model Diagram – Construct

For the sake of this document we have used the above diagram as a basis to illustrate the "V" Model. It consists of shapes, arrows, and terminology - this structure will be used to explain the underlying principles of the "V" model.

Circles - At the upper left and upper right corner of the diagram are two green circles – they are used to denote the origin and completion of a project. The "V" model does not address the factors or activities an organization utilizes to authorize the startup of a "project" nor does it address the procedures or organizational infrastructure required to support an application once it is developed and made available in a production environment.

Rectangles – Seven rectangles are identified on the diagram. The generic terms (Requirements Definition, High Level Design, Detail Design, Coding, Unit Testing, Integration Testing, Acceptance Testing) have been used to reflect the phase names that are applied by a number of industry recognized methodologies. The "V" model does not suggest, imply or demand the terms an organization uses to define its development process, phases or methodology (i.e, an organization using the term "Preliminary Analysis" as its initial phase to define requirements would use that term rather than "Requirements Definition" depicted in the illustrated "V" model).

Diagonal Arrows – Two diagonal arrows are used to distinguish the flow of a project. One arrow originates at the top left (Project Startup) and flows through to the coding phase of the project – this arrow reflects the *development* portion of the model. The other arrow originates at coding phase of the project and flows through to the project being delivered to the maintenance and support team – this arrow reflects the *testing* portion of the model.

Horizontal Arrows – Three horizontal arrows are used to illustrate that there must be a correlation between the *development* portion (requirements definition and design) of the model and the *testing* portion that has to be performed to verify that the application being built reflects the authorized requirements and design. These horizontal lines have been labeled with "Review/Test".

"V" Model – Principles

The following principles are inherent when the "V" model is applied.

Large to Small - This principle states requirements, standards, testing from a hierarchical perspective. For example, requirements (left side of the diagram) are identified and defined as a project team evolves through the Requirements Definition, High-Level Design, and Detailed Design phases of their project. As each of these phases are completed the requirements they are defining become more and more refined and detailed (when defining the requirements to build a space shuttle a *requirement* defined during the Requirements Analysis phase may be that the space shuttle required landing gear whereas a *requirement* defined at the Detailed Design phase would be that the wheels of the landing gear are to be made of rubber and able to withstand the force of landing at 300 mph – the requirements get further refined with additional granularity as the project evolves).

Data/Process Integrity – This principle states that the successful design of any solution requires the incorporation and cohesion of both data and process(es). As the requirements are defined data and process elements must be identified for each and every requirement.

Scalability – This principle states that the “V” concept has the flexibility to accommodate any IT project irrespective of its size, complexity or duration. The “V” concept is as applicable to a large mainframe development project applying a waterfall approach as it is to a web-based development project applying agile techniques.

Cross Referencing – This principle states that there must be a direct correlation between every requirement that has been defined with a corresponding and verifiable testing activity and result that substantiates that each and every authorized requirement has been incorporated into the completed application.

Tangible Documentation – This principle states that there must be tangible documentation (electronic and/or hardcopy) created as the project evolves. This documentation is required and applied by both the development project team and the support team that will be maintaining the application once it is available in a production environment. The “V” model does not suggest specific document titles or templates or formats. The “V” model does not suggest how many documents must be prepared or authorized or utilized throughout the projects life.

“V” Model Flow - Seven Steps

Step I - At this level (Requirements Definition and Acceptance Testing) the project team is accountable for three primary responsibilities. The first responsibility is to begin defining the high level (most broad) requirements of the application being developed. The second responsibility is to begin planning the testing activities that will have to be performed to verify the high level requirements have been incorporated and satisfied. The third responsibility is to establish the pre-defined conditions that will have to be tested to verify the high level requirements (most broad) have been incorporated and satisfied.

Step II - At this level (High Level Design and Integration Testing) the project team is accountable for four primary responsibilities. The first responsibility is to further refine the granularity of the high-level requirements established during the Requirements Definition phase. The second responsibility is to begin creating a high level solution design based on the requirements established during the Requirements Definition. The third responsibility is to begin planning the testing activities to be performed to verify the requirements (at the High-Level Design phase) have been incorporated and satisfied. The fourth responsibility is to establish the pre-defined conditions that will have to be tested to verify the High-Level Design phase requirements have been incorporated and satisfied.

Step III - At this level (Detailed Design and Unit Testing) the project team is accountable for four primary responsibilities. The first responsibility is to further refine the granularity of the requirements established during the High-Level Design phase. The second responsibility is to continue refining the design and solution based on the requirements established during the High-Level Design phase – this includes the creation of specifications (functional and/or technical) used to create the application. The third responsibility is to begin planning the testing activities to be performed to verify the requirements (at the Detailed Design phase) have been incorporated and satisfied. The fourth responsibility is to establish the

pre-defined conditions that will have to be tested to verify the Detailed Design phase requirements have been incorporated and satisfied.

Step IV – At this level (Coding) the project team has one primary responsibility. That responsibility is to translate the specifications created in the Detailed Design phase into technical code (in whatever platform or language).

Step V – At this level (Unit Testing) the project team is accountable for three primary responsibilities. First, to execute the Unit Test phase activities according to pre-defined Unit Test plan (established in Step III). Second, to identify and document deviations between “pre-defined anticipated results” with “actual testing results” for each and every unit/program. Third, to ensure all pre-defined Unit Test cases have been executed and all the expectant results have been achieved. There will be one or several iterations of this step between the development team and the testing team to ensure all of the appropriate requirements have been defined and successfully tested – once this step has been finalized the project team will continue with Step VI.

Step VI – At this level (Integration Testing) the project team is accountable for three primary responsibilities. First, to execute the Integration Test phase testing activities according to plan (established in Step II). Second, to identify and document deviations between “pre-defined anticipated results” with “actual testing results” for each and every sub-system. Third, to ensure all pre-defined Integration Test cases have been executed and all the expectant results have been achieved. There may be one or several iterations of this step between the development team and the testing team to ensure all of the appropriate requirements have been defined and successfully tested – once this step has been finalized the project team will continue with Step VII.

Step VII – At this level (Acceptance Testing) the project team is accountable for three primary responsibilities. First, to execute the Acceptance Test phase testing activities according to plan (established in Step I). Second, to identify and document deviations between “pre-defined anticipated results” with “actual testing results” for the application. Third, to ensure all pre-defined Integration Test cases have been executed and all the expectant results have been achieved. There may be one or several iterations of this step between the development team and the testing team to ensure all of the appropriate requirements have been defined and successfully tested – once this step has been finalized the project team will have completed its work and the application can be made available in the production environment.

“V” Model Diagram – Benefits

Applicability – the “V” model affords organizations and project teams a guide to performing and completing projects in a consistent and repeatable manner. Applying the principles of the “V” model ensures the user requirements are identified and documented, the authorized requirements can be traced into the functions of the completed application, and the application reflects the user requirements.

Flexibility – the principles of the “V” model are applicable in both development and maintenance/support environments and can be applied using one or many (spiral, rapid application development, prototyping, waterfall, agile) approaches.

Formality/Process – in applying the principles of the “V” model an organization can establish a formal and standardized process they use to develop and/or maintain applications. Having a standardized process enables them to quantify the quality being delivered by the process, establish and leverage process metrics to continually evaluate and improve the process, increase versatility of staff to work on varied applications, reduce training costs by limiting the number of lifecycles, methodologies, deliverables being used by multiple application teams.

Support Documentation – there is always a balance that must be considered when developing a new application. The equation – the time saved to create the application by accelerating the development process versus the time lost in trying to find information to maintain the same application without effective reference material and documentation. Every organization is unique as are the environments, methods, tools, and techniques they use to develop and maintain applications – the amount of documentation needed is subjective to the organization. The “V” model provides a logical and practical framework to ensure the appropriate amount of documentation can be created during development and referenced in support.

Adherence – all of the principles of the “V” model can be applied using the majority of all industry recognized methodologies, lifecycles and project management tools.

QAassist is the industry recognized benchmark in information technology (IT) methodologies for small and mid-sized business (SMB's) – including the certification and support of practitioners delivering QAassist IT Methodology solutions. Article authored by Cameron Watson – President of QAassist. Visit [QAassist's website—www.qaassist.com](http://www.qaassist.com)