



Promoting IT Efficiency

QAIassist
IT Methodology – General Context

IT Methodology – General Context

From the inception of Information Technology (IT), organizations and people have been on a constant quest to optimize the evolving technologies. IT resources continue to strive to find ways to deliver systems and applications more efficiently, cheaper, of better quality and reliability.

To contribute to the quest, various methodologies have been devised, best practices have been learned and applied, life-cycles have been created, and deliverables have been customized to address specific project needs. Through this evolution new terminology continues to be introduced, additional buzzwords are continually used, and new technical semantics are routinely adopted.

When the discussion turns toward methodology, there are so many terms and so many interpretations that the ultimate result is a lack of clarity and much misunderstanding. What is deemed a "methodology" at one organization is deemed a "life-cycle" at another organization. What is deemed a "delivery process" at one organization is called a "methodology" at another organization. Suffice it to say, there is a great deal of time and effort that must be applied to calibrate a context for discussing IT methodologies.

Though the number of definitions used to describe a "methodology" is as varied as the number of organizations applying it, the easiest and most proficient way to discuss the term is to recognize it can be used as a ***noun*** or a ***verb***. In discussing methodology from these two (noun versus verb) perspectives enables a clearer context to be defined and appreciated.

IT Methodology - (as a Noun)

As a noun, "methodology" is best described as a process or framework with a beginning and an end. Between the beginning and end there is predefined and preset criteria. In some cases the predefined criteria is stages, iterations or phases, in other cases it may be work products to be completed, and in other cases it may be a specific deliverables. It and its pre-defined criteria act as a framework to guide the path (required work products and deliverables to be completed) and activities (the verb) needed to complete an IT project.

In the "noun" context, specific methodologies have been developed as products and are available in the marketplace. Each methodology relies on a specific pre-defined set of criteria (phases, work products, deliverables) that acts as the framework project teams utilize to guide projects from inception through completion. These methodologies vary in scope and applicability - the scope ranging from project management, to software development, to software testing. Examples of the methodologies include

ITIL

The Information Technology Infrastructure Library (ITIL) is a collection of best practices that aim to improve and then maintain a certain level of computing services quality in the information technology sector. ITIL covers organizational structure and skill requirements for an IT organization via a comprehensive set of procedures with which an organization can manage its IT operations. These procedures do not rely on a particular vendor's technology and apply to all aspects of IT infrastructure.

ITIL consists of a collection of 7 books. The sets are sub-divided into disciplines, each of which is focused on a specific subject.

1. Service Delivery: Covers the processes required for the planning and delivery of quality IT services, and looks at the longer-term processes associated with improving the quality of IT services delivered.

1. IT Financial Management
2. Capacity Management
3. Availability Management
4. IT Continuity Management
5. Service Level Management

2. Service Support: Describes the processes associated with the day-to-day support and maintenance activities involved in the provision of IT services.

1. Change Management
2. Release Management
3. Problem Management
4. Incident Management
5. Configuration Management
6. Service Desk

3. Planning to Implement Service Management: Examines the issues and tasks involved in planning, implementing, and improving service management processes within an organization; also addresses the issues associated with addressing cultural and organizational change, the development of a vision and strategy, and the most appropriate method of approach.

4. Security Management: Details the process of planning and managing a defined level of security for information and IT services, including all aspects of security incidents. Also includes the assessment and management of risks and vulnerabilities and the implementation of cost-justifiable countermeasures.

5. Information and Communications Technology (ICT) Infrastructure Management: Covers all aspects of ICT infrastructure management from identification of business requirements through the procurement process, to the testing, installation, deployment, and ongoing operation and optimization of ICT components and IT services.

1. Network service Management
2. Operations Management
3. Management of local processors
4. Computer installation and acceptance
5. Systems Management

6. The Business Perspective: Provides advice and guidance to help IT personnel understand how they can contribute to business objectives and how their roles and services can be better aligned and exploited to maximize contribution.

7. Application Management: Describes how to manage applications from the initial business need through all stages in the application lifecycle, up to and including retirement. Places emphasis on ensuring that IT projects and strategies are tightly aligned with those of the business through the application lifecycle, to ensure that business obtains the best value from its investment.

From the beginning, the ITIL framework has been publicly available (however, it is copyright protected). This means that any organization can use the framework described by the OGC in its numerous books. Because of this, ITIL guidance has been used by a wide range of organizations including government, energy, public utilities, retail, finance, and manufacturing. Very large organizations, very small organizations and everything in between have implemented ITIL processes.

Prince2

PRINCE2 is project management methodology that takes a process-based approach. It consists of the following 8 high-level processes:

1. Directing a Project:

This process defines the responsibilities of the Project Board for the project. The project manager keeps the Project Board informed with regular reports. The board otherwise leave the day-to day-management of the project to the Project Manager.

2. Planning:

Planning is a process involved throughout the project's life-cycle.

3. Starting up a Project:

The purpose of this process is to ensure that project is set up correctly. It is a pre-project process that which determines if the project would be worthwhile and viable. If it is, then it makes sense to seek commitment of resources.

4. Initiating a Project:

In order for a project to be approved it must be carefully planned to show how it will meet its goals. Once planned, the project must be approved by the Project Board before implementation can commence.

5. Controlling a Stage:

PRINCE2 projects are divided into stages so a project can be more easily managed and controlled. This process covers the day-to-day management of the project by the Project Manager.

6. Managing Product Delivery:

PRINCE2 is a product based system. In this context, a product can be a physical item like a book or it can be an intangible such as a service agreement. This process creates the products of the project and is where most of its resources are used.

7. Managing Stage Boundaries:

According to PRINCE2 principles, each stage must be completed and approved by the project board before the go ahead is given to proceed to the next stage.

8. Closing a Project:

Another principle of PRINCE2 is that projects must be closed down in a controlled and orderly way. This involves evaluating the project's results. Any lessons learned are recorded, a handover document is created if necessary, and a post implementation review is planned.

Rational Unified Process (RUP)

IBM's Rational Unified Process® (RUP®) is a comprehensive process framework that provides industry-tested practices for software and systems delivery and implementation and for effective project management. It is one of many processes contained within the Rational

Process Library, which offers best practices guidance suited to your particular development or project need.

The RUP process framework provides:

- Processes based on the best practices adopted in thousands of projects worldwide. Avoid inventing everything from scratch and reuse processes that have been successful for other organizations.
- Capability patterns that allow project managers to rapidly add or remove reusable chunks of processes addressing common problems. Because no two projects are alike, project managers can modify the process to address specific project needs.
- Ready-to-use delivery processes to provide the project manager with a quick starting point for planning and initiating a project. A delivery process provides an initial project template and identifies which type milestones to use in the project, which work products to deliver by each milestone, and which resources are needed for each phase.

RUP promotes iterative development and organizes the development of software and systems into four phases, each consisting of one or more executable iterations of the software at that stage of development.

QAIassist Integrated Methodology

A deliverable based methodology that integrates the disciplines of three (project management, Software development, software testing) methodologies. The methodology is scaleable for small, medium and large sized projects and can be utilized within a multitude of delivery model (waterfall, agile, prototyping, etc) environments.

The methodology has been designed in accordance with industry recognized standards and best practices and can be customized to meet the specific requirements of any project - it can be implemented in whole or in part.

The methodology provides

- Business Stakeholders are able to establish project definition and provide oversight for applications being developed - business community has ownership and participates on project teams from inception
- IT Management is able to rely on a repeatable delivery mechanism (development/maintenance) - ensuring consistency in the completion, cost and quality of applications and maintenance being delivered
- Formal deliverables (scaled to the scope and complexity of each project) are developed and authorized - ensuring consistent communication exists between all project team (business and technical) members

- Business Analysts are able to contribute to ensure their business requirements are being built into the solution
- Project Managers are able to identify and customize the creation and approval of deliverables based on the unique characteristics (scaleable) of the project - ensuring a balance is struck between schedule, cost and quality
- Project Team members (Business Analysts, Technical Architects, System Analysts, Programmers, Testers) can be assigned specific deliverables - making progress, issues and risks become more manageable
- Versatility to adapt to multiple (prototyping, agile, waterfall) development and maintenance environments -rework (development & maintenance environments) is significantly reduced
- Organizational Quality Assurance/ Governance resources are able to utilize the methodology as a benchmark to establish, evaluate and communicate IT performance metrics and measurements
- Repeatable delivery mechanism provides greater flexibility in the deployment of IT resources - they can be utilized in development or maintenance environments and across multiple applications

IT Methodology - (as a Verb)

As a verb, "methodology" can best be described as being the activities that must be completed to create and satisfy the predefined criteria of a methodology (the noun) - in essence, the methodology (noun) delivery model. Project teams members perform activities to create work products and deliverables within specific stages, iterations or phases as part of the delivery model.

In the "verb" context, the specific activities and delivery approaches needed to complete the work products and deliverables have been defined and are available in the workplace. Project teams perform the activities through which a product or deliverable is created - the intent is to define the path a project team must utilize through the methodology (noun) to complete their project. Examples of these methodology delivery models include

Waterfall

The waterfall delivery model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is rigid and linear. Waterfall development has distinct goals for each phase of development where each phase is completed for the next one is started and there is no turning back.

The perceived advantages of the waterfall process is that it allows for departmentalization and managerial control. A schedule is typically set with deadlines for each stage of development and a product can proceed through the development process. In theory, this process leads to the project being delivered on time because each phase has been planned in detail.

In practice, waterfall development often falls short of expectations as it does not embrace the inevitable changes and revisions that become necessary with most projects. Once an application is in the testing stage, it is very difficult to go back and change something that was not thought of in the concept stage. Alternatives to the waterfall model include joint application development (JAD), rapid application development (RAD), sync and stabilize, build and fix, and the spiral model.

Spiral

The spiral delivery model extends the waterfall model by introducing prototyping. It is generally chosen over the waterfall approach for large, expensive, and complicated projects.

At a high-level, the steps in the spiral model are as follows:

1. The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
2. A preliminary design is created for the new system.
3. A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
4. A second prototype is evolved using four steps:

1. Evaluate the first prototype and identify its strengths, weaknesses, and risks.
2. Define the requirements of the second prototype.
3. Plan and design the second prototype.
4. Construct and test the second prototype.
5. At the project sponsor's option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could result in a less-than-satisfactory final product.
6. The existing prototype is evaluated in the same manner as was the previous prototype, and, if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
7. The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
8. The final system is constructed, based on the refined prototype.
9. The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.

Rapid Application Development

The rapid application development (RAD) delivery model proposes that products can be developed faster and of higher quality delivered by the traditional waterfall model:

- * Using workshops or focus groups to gather requirements.
- * Prototyping and user testing of designs.
- * Re-using software components.
- * Following a schedule that defers design improvements to the next product version.
- * Keeping review meetings and other team communication informal.

There are commercial products that include requirements gathering tools, prototyping tools, software development environments such as those for the Java platform, groupware for communication among development members, and testing tools. RAD usually embraces object-oriented programming methodology, which inherently fosters software re-use. The most popular object-oriented programming languages, C++ and Java, are offered in visual programming packages often described as providing rapid application development.

Agile

The Agile delivery model is a conceptual framework for undertaking software engineering projects. There are a number of agile software development methodologies

Most agile methods attempt to minimize risk by developing software in short timeboxes, called iterations, which typically last one to four weeks. Each iteration is like a miniature software project of its own, and includes all the tasks necessary to release the mini-increment of new functionality: planning, requirements analysis, design, coding, testing, and documentation. While an iteration may not add enough functionality to warrant releasing the product, an agile software project intends to be capable of releasing new software at the end of every iteration. At the end of each iteration, the team reevaluates project priorities.

Agile methods emphasize realtime communication, preferably face-to-face, over written documents. Most agile teams are located in a bullpen and include all the people necessary to

finish the software. At a minimum, this includes programmers and the people who define the product such as product managers, business analysts, or actual customers. The bullpen may also include testers, interface designers, technical writers, and management.

Agile methods also emphasize working software as the primary measure of progress. Combined with the preference for face-to-face communication, agile methods produce very little written documentation relative to other methods - this lack of system documentation and reference often limits the efficiency of IT resources responsible for updating, maintaining or supporting an application once it has been placed in the production environment.

Joint Application Development (JAD)

The Joint Application Development (JAD) delivery model aims to involve the client in the design and development of an application. This is accomplished through a series of collaborative workshops called JAD sessions. Two employees of IBM, Chuck Morris and Tony Crawford, developed the JAD methodology in the late 1970s and began teaching the approach in to the 1980s.

In contrast to the Waterfall approach, JAD is thought to lead to shorter development times and greater client satisfaction, both of which stem from the constant involvement of the client throughout the development process. On the other hand, with the traditional approach to systems development, the developer investigates the system requirements and develops an application, with client input consisting of a series of interviews.

Rapid application development (RAD), a variation on JAD, attempts to create an application more quickly through strategies that include fewer formal methodologies and reusing software components.

Scrum

Scrum is an agile delivery model used for project management. Its goal is to dramatically improve productivity in teams previously paralyzed by heavier, process-laden methodologies. Its intended use is for management of software development projects as well as a wrapper to other software development methodologies such as Extreme Programming.

Scrum is characterized by:

- * A living backlog of prioritized work to be done.
- * Completion of a largely fixed set of backlog items in a series of short iterations or sprints.
- * A brief daily meeting (called a scrum), at which progress is explained, upcoming work is described, and obstacles are raised.
- * A brief planning session in which the backlog items for the sprint will be defined.
- * A brief heartbeat retrospective, at which all team members reflect about the past sprint.

Scrum is facilitated by a scrum master, whose primary job is to remove impediments to the ability of the team to deliver the sprint goal. The scrum master is not the leader of the team (as

they are self-organizing) but acts as a productivity buffer between the team and any destabilizing influences.

Scrum enables the creation of self-organizing teams by encouraging verbal communication across all team members and across all disciplines that are involved in the project. A key principle of scrum is its recognition that fundamentally empirical challenges cannot be addressed successfully in a traditional "process control" manner. As such, scrum adopts an empirical approach – accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to respond in an agile manner to emerging challenges.

*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. Visit [QAassist's website](http://www.qaassist.com)—
www.qaassist.com*