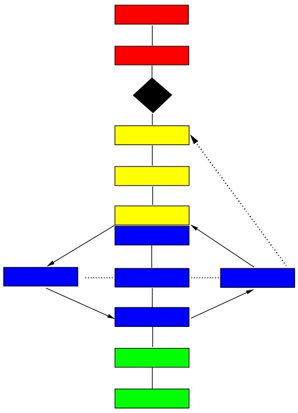
Application Development Projects Guide

MITP
v5.1



Document Number MICG1ADP

Edition Notice
First Edition (September 1995)

This edition applies to Version C5.0 of Managing the Implementation of the Total Project (MITP), and to all subsequent releases and modifications until otherwise indicated in new editions.

A form for reader's comments appears at the back of this document. If the form has been removed, address your comments to:

Allturn Group International B.V.
Groenendal 7c
5405 AS Uden (NB) The Netherlands
Email: Info@AllturnGroup.com

Phone: 0031 (0) 6 20 35 67 51

When you send information to Allturn Group International, you grant Allturn Group International a nonexclusive right to use or distribute the information in any way appropriate without incurring any obligation to you.

Table of Contents

Figures

PREFACE About This Document

This document describes the special considerations for Application Development (AD) projects within an MITP framework. It gives an overview of how MITP is used in application development (AD) projects and introduces some of the more important points relevant to these projects.

For information about the MITP life cycle, the key techniques and the support techniques, see the MITP Handbook. A glossary of terms can be found at the back of the MITP Handbook.

Who Should Read This Document

This document is intended for Project Managers, although other people can read it and extract useful information from it.

How to Use This Document

This document is organized in a sequence of topics appropriate to the MITP methodology, following the sequence of the MITP life cycle:

- Identifying the Project
  - Identification
  - Endorsement
- Establishing the Project
  - Project definition
  - Agree Organization and Responsibilities
  - Start-up - plan and build teams

  Managing the Project
  - Plan
  - Track progress
  - Review
  - Manage exceptions
- Completing the Project
  - Completion
  - Benefits

In addition, the Table of Contents gives a full overview of the topics covered.

ISO9000 Control Information

The owner of all MITP Version C5.0 material is ALLTURN GROUP INTERNATIONAL. The MITP License applies to the current version only. Future revisions, which are under version number control, may be made available under upgrade license terms from ALLTURN GROUP INTERNATIONAL. The current license does not cover upgrades.

1 Introduction

The basic aim of using MITP for Application Development (AD) projects is to maximize the number of AD projects that complete successfully. A successful project is defined as one that meets the expectations of those involved with or affected by the project. This normally includes:

- Those who use the new or changed system, either directly, such as through an on-line terminal, or indirectly, such as by the receipt and use of a printed report from the system
- Those who operate the system
- Those who are responsible for maintaining and enhancing the system in the future
- Those who work on the project
- The sponsors and those expecting business gains from their investments.

Subtopics
1.1 Project Type
1.2 Project Size

1.1 Project Type

The information in this document is intended for AD projects, where the major objective is to develop or amend a computer application system. The project itself could be a part or subproject of a larger project.

You use MITP as the general approach to project management but this document focuses on the specific needs of AD projects.

1.2 Project Size

You can apply the principles of MITP to AD projects of any size. Indeed, one of the principles of MITP is that procedures should vary according to the size and risk of a project. However, MITP is about managing projects. There is some work in an AD department that it is inappropriate to treat as a project. In particular, small changes (including error corrections) to live systems is usually best managed as a continuous process under a line manager, rather than a series of projects. It is recommended that the majority of maintenance and enhancement work is packaged into releases each of which can be managed as a project under MITP. Any individual item of work over a particular size should also be managed as a project.

2 What Makes AD Projects Different?

AD projects have aspects that distinguish them from other projects in several areas including:

Unknown product
Very often in an AD project the work being done is new to the people or organization. This means that previous experience cannot be used as reliably or effectively to plan or manage the project. Also software tends to be less visible than other products.

Complex product
The complexity of an AD project can be significant. There may be many subprojects involved, each with interdependencies. The resulting application may have to interface with existing products or other functions. If the application being implemented is a new application then far-reaching cultural changes are often required. There is also quite often an element of the project which requires the use of leading-edge technology to prepare for the implementation of an application. All require successful completion in order that the total project is implemented to meet end user needs.

Requirements
The gathering and documenting of correct requirements is absolutely fundamental. Without this being successfully completed, the project will almost certainly not meet its goals. Detailed requirements gathering for AD projects, especially where new technology is involved is often also a difficult task. Users tend to express needs as complaints against existing systems, which although useful input, may not give a definition of what could deliver business gains. As requirements gathering can sometimes be more of an art than a science, the subject is best covered by the specialist material available, rather than here.

People-intensive
The process is very dependent on people and team skills, given that many projects tend to be 'new' in many significant areas. If the right skills and teams are built the project stands a better chance of success regardless of other factors. If this is not the case then, whatever the quality of the other dependencies, the chance of success is considerably diminished.

Quality management
Testing is traditionally thought of as the system testing of the final product. However, quality validation is vital throughout the process from start to finish. The earlier in the process that errors are eliminated, the less it costs the project. Because of this, a rigorous quality plan is required to be in place and implemented from the start of the project.

Estimates
Estimating is performed for identified parts of the project. If the scope is not fully understood and tasks missed from the development, quality, or management processes, or there is some hidden enabling work to be performed, the basis for estimating is incomplete and the resulting estimates are inaccurate. Typically, for example, test data availability, validation or generation, interfaces, and the need for migration or conversion facilities get overlooked.

Effects of change
Although change is inevitable in any project, an AD project needs to limit and plan for the amount of change to be allowed. If this change allowance is exceeded then a process of escalation should be invoked to decide the impact of further change. In addition, if one is a significant way through the project, there is less chance of implementing change without major impact.

Progress
This can seem to be invisible until late in the project. It can potentially give a false picture as it may be very difficult to monitor true progress and correct where necessary (the 90% complete syndrome, where tasks are deemed to be nearly complete by the programmer every time that the status is ascertained). Also, it can be difficult for you to demonstrate to the sponsor or investor that the project is on track. Software is not like a building. The outside observer cannot see each floor being erected as time goes by. If possible, the project should be organized to be able to show demonstrable results at regular intervals.

3 Identification and Endorsement

This topic covers the things you need to consider during Phase 1 of a MITP for AD project.

Subtopics

3.1 Identification

3.2 Endorsement

3.1 Identification

MITP for AD assumes that the decision on which projects should be undertaken has already taken place. There will have been a request evaluation and project selection process.

3.2 Endorsement

AD projects generally demand increased emphasis on the following areas as part of the Endorsement activity.

3.2.1 Identify Key People

In MITP, project organization covers two areas: the organization and responsibilities of the people, and the organization, or structure, of the work. MITP recognizes that organizations have differing line management organizations, and that individual projects differ in terms of size, risk, and breadth; that is, the number of parts of the organization involved in, or affected by the project, and the people and skills available to fill management positions on the project. There is no one correct project management organization for all projects.

This topic suggests a basic template for an AD project organization and recommends job specifications for the positions in the organizations.

The template should form a starting point for discussion on what should be the organization for a particular AD project, accepting that certain aspects of the organization are mandatory, for example, that there should be a Project Manager and Subproject Managers. Typically, the AD project has its own Project Manager, reporting to the Project Manager of the business project. However, the ultimate decision should be in the light of the specific characteristics of the individual project.

Figure 1 shows the project reporting through a single Project Manager to a Project Sponsor. You may have one or more subprojects, each of which has a project leader who reports to you. The importance of communications with the eventual users of the IS system being built or amended should be emphasized. To this end, the project organization includes a requirements group that represents the requirements of the user departments, including aspects such as implementation plans as well as the function of the system. The term user embraces all who are affected by the system including the operations department. The user project leader co-ordinates the views of the requirements group and also manages the user work on the project.

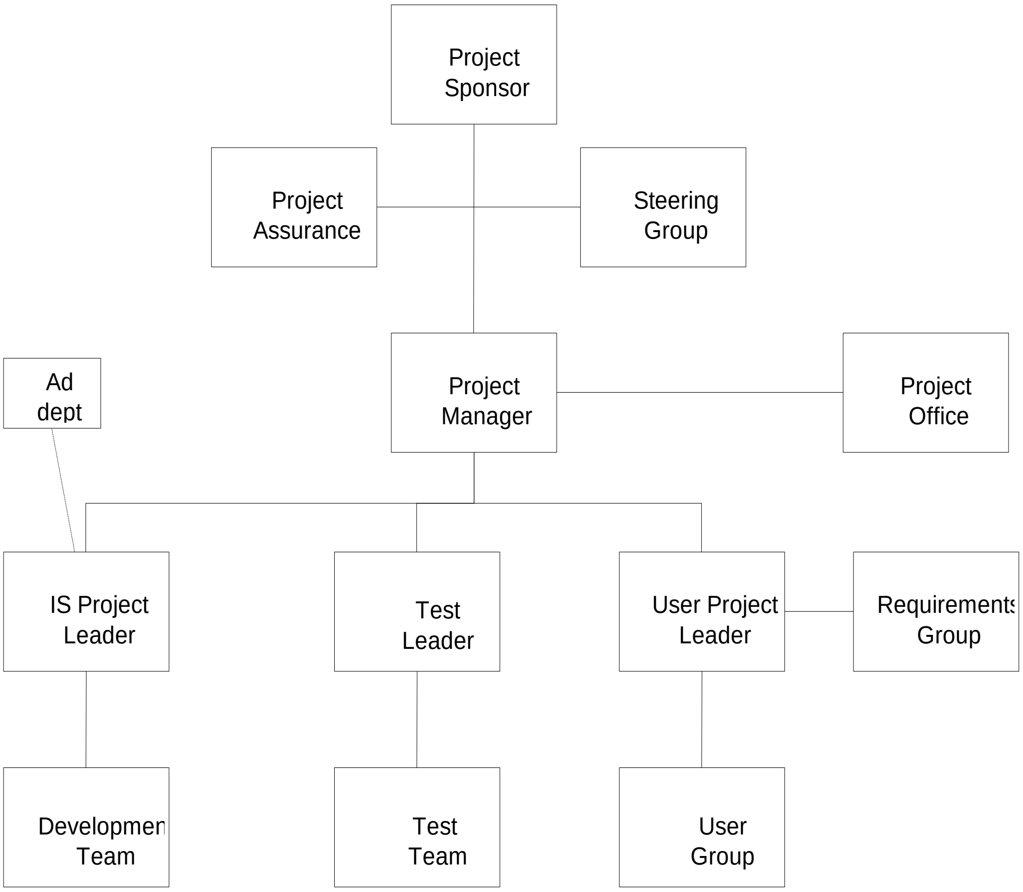


Figure 1. AD Project Organization Template

The roles and responsibilities of these people are more fully described in "Organization" in topic 5.0 and in Organization and People Management Guide.

3.2.2 Confirm Feasibility

Before project endorsement, a feasibility study should have been performed (for some small projects it may be a very brief study). Project endorsement may take place some time after the feasibility study and may involve different people. The purpose of this activity is to ensure that the feasibility study findings are still valid and cover (at least):

- Outline of recommended solution
- Proposed development approach and phase structure
- Estimated costs and timescales
- Estimated benefits
- Risk assessment
- Outline of alternative solutions considered
- Reason for solution, if tactical and not strategic.

and is agreed between the nominated users and developers.

In some situations the feasibility study may have to be performed again, completely or in part, to establish the above findings.

You review the feasibility study report to determine if it is complete and still relevant. Rework may be required. In particular the estimated schedule is reviewed to ensure it represents the best current estimate. You can use rules of thumb for this validation. Where the project is already well defined an outline task estimate for the first phase should be produced.

The risk assessment presents a broad categorization of the project into high-, average-, or low-risk. In addition to the MITP risk process, risks are assessed for AD projects by considering:

- Elapsed timescale of project
- Complexity of requirement
- Functional implications
- Conversion required
- User team and solution developer experience
- Project dependencies
- Number of departments and locations involved in providing solution
- Completeness of project costs.

You can use a risk assessment checklist to help identify the risks. For each risk, you must identify what could be done to eliminate or minimize it.

The business case should be documented from all of the information gathered, and should cover:
Development costs (human resources including training, machine, buildings, networks, travel, and other)
Running costs (hardware, software, people, buildings, and other)
Benefits (headcount savings, machine saving, business benefits).

The above should be confirmed with the Project Sponsor.

3.2.3 Project Assurance

You are strongly recommended to establish a project assurance function to provide an independent view of plans, status, and deliverables for key projects. Such a function also provides a centre of competence for project management, is responsible for project management education, and maintaining project management standards and guidelines.

It is often the case that those closely associated with a project fail to notice problems, or do not see all possible solutions to a problem. Project assurance can provide an extra perspective on a project and thus be an invaluable aid to management.

Register your project with Project Assurance. Many organizations assign a reference number at this time in the appropriate project and cost control systems.

4 Definition

AD projects generally demand increased emphasis on the following areas within the Definition activity of MITP.

Subtopics

4.1 Milestones

4.2 Additional Project Definition Workshop Topics

4.3 Joint Application Design

4.1 Milestones

AD project milestones can be based on the AD method employed. Designate some as key review points, when the project will be reviewed and plans updated. Normally include development phase ends.

MITP does not explicitly define the tasks to be performed on a project. It also does not define in detail the phase structure that a project should follow, although it does indicate the characteristics that a phase structure should have. This means that MITP can coexist with, and support, any development method.

The organization of the work on a project represents an important element in controlling the project and the work can be structured in two major ways:

- By the nature of the work (subprojects)
- By timescale (milestones and phases).

A subproject is a piece of work that has clear objectives, has minimal overlap with other subprojects, and can sensibly managed by a single project leader. Examples of subprojects include:

- AD
- User training
- Equipment installation.

Each subproject breaks its work down into intermediate objectives represented by milestones. For the AD subproject, the milestones includes (but may not be restricted to) the end of development phases. Depending on the development method being used, these phases may have different names. For traditional waterfall development methods, the phasing reflects the basic development structure:

- Requirements
- Design
- Build
- Test
- Implement.

For incremental development with the rapid development techniques of prototyping, small teams and iteration, the development structure might be:

- Requirements
- High-level design and build
- Full design and build
- Test
- Implement.

The actual phases and milestones to be used on a project are defined during project definition, and vary from one project to another, depending on factors such as size and the development approach to be taken. Certain milestones are designated as key major review points in the project, and these usually include development phase ends. At key milestones the project is reviewed in its entirety, and project plans are formally updated to plan the next phase. Key milestones thus break the project down into manageable pieces and, together with the subproject concept, provide for delegation of responsibilities, and clear objectives for project teams.

One form of AD method involves Control Points, which act as milestones within the project's progress. Such a method would give you:

- An Initiation Control Point (ICP): confirmation that the AD project is defined within the context of a business project and MITP controls and methods are in place
- A Continuation Control Point (CCP): the business and data models are defined, the requirements gathered, a Joint Application Design (JAD) or Joint Application Requirements (JAR) workshop held, an outline system design is present, service level objectives are defined, and programming can commence (see "Joint Application Design" in topic 4.3)
- A Release Control Point (RCP): the programs are developed and tested, a Pilot or User Acceptance test is completed, roll-out and implementation plans are in place: the product can be released into production

4.2 Additional Project Definition Workshop Topics

The following are suggested additions to your list of topics for a Project Definition Workshop (PDW) AD projects.

4.2.1 Metrics

Information to be captured for future use Data from projects is a valuable asset if properly captured, analysed, and used. Project data can be used for:

- Updating standards, procedures, and guidelines
- Building and updating estimating models and matrixes
- Measuring product quality
- Measuring developer productivity
- Measuring the effects of introducing new methods and tools
- Measuring user satisfaction.

Metrics to be captured during this project and the capture process defined in the Project Definition Where there is an already agreed and documented standard set of metrics and data capture processes, it is only necessary to highlight any deviations from the standard set.

The following is a suggested set of project metrics (many involve comparisons of planned versus actual):

- Project type
- Project size category
- Hardware platforms and their percentages
- Risk assessment history
- Key dates
- Size and complexity:
  - Function points
  - Thousands of lines of code, by language used and type (new, changed, deleted etc.)

- Project changes handling (net hours) covering logging, investigating, impacting, accepting, rejecting, and implementing
- Project issues and problem handling (net hours) covering logging, investigating, impacting
- Project actions (net hours) covering the actions necessary to implement the resolutions of issues and problems
- Project error handling (correction rework) (net hours) covering logging, investigating, impacting, and fixing.
- Include only correction rework for completed tasks. Only include correction rework where the error originated in a previous work phase. Correction within a work phase should be handled within the planned task allowance or by charging the rework to contingency.
- Contingency (net hours).
- Project effort by project phase.
- Cost of quality (quality planning, appraisal effort, failure correction effort etc.).

4.2.2 Technical Environment

This is the probable technical environment for the development, testing, installation, and production of the applications need to be stated.

This should include the probable hardware environment, software mix, languages, methods, and tools. If these are not all already available and in place, this needs to be highlighted at this point as approval for their acquisition and installation would need to be agreed in principle at this stage.

4.2.3 Facilities

Any facilities that are required during the development project or during implementation of the resulting product need to be identified and documented during project definition. This should enable the supply of the facilities to be planned and budgeted for in good time.

4.2.4 Service Levels

Any non-standard service level requirements need to be specifically stated, agreed, and documented (for example, 24 hour non-interrupted availability, disaster fallback etc.).

4.2.5 Quality Plan

The quality plan is specific to a project and not a standard procedure. Reference to a standard quality procedure should be made to avoid duplication for each project. Only project-specific and approved deviations from standard quality procedures should be included. The level of detail defined within the project quality plans should be appropriate to the size, complexity and risk assessment of the project. The primary purpose of a quality plan is to define clearly the product quality completion criteria and how the project team measures and demonstrates that this can be achieved, as work progresses through each project phase. A secondary purpose is to define targets for process quality improvement and define how the project team can improve the AD process methods and tools and assess the improvements made.

The purpose of the quality plan is to provide specific focus on quality within a project. It defines the objectives, methods, resource, and measurements of quality in the project. It is expanded for each stage of development as more detail is available and individual work items are identified. It is updated with actuals against plan and modified according to results.

The project plan needs to reflect the tasks and checkpoints required by the quality plan. Changes in the quality plan may require changes in the project plan (and the other way around). Formal approval of quality plan changes should be required.

The following information should be included:

- Quality objectives:
  - Product quality requirements
  - Market-driven quality improvement targets.
- Responsibilities
- References to supporting material:
  - Standards and practices
  - Tools, techniques, and methods
  - Test plan.
- Outline quality plan:
  - Development approach
  - Verification approach
  - Stage quality targets
  - Expected defect detection effectiveness
  - Actions to be taken for deviations from plan
  - Criteria for revising and approving the quality plan
  - Product assurance review schedule.
- Stage quality plan:
  - Identified work items (deliverables)
  - Identified work item quality checks
  - Target defect density (by work item by stage)
  - Detailed inspection, simulation, test plans
  - Plan and actual tracking
  - Risk assessment.

4.3 The Joint Application Design (JAD) workshop

The Joint Application Design (JAD) workshop can be a useful vehicle for the planning, designing and requirements definition of an AD project. An overview of JAD is provided below, with more detail available in specific JAD documentation.

4.3.1 JAD Types

There are two types of JADs that can be performed on a project:

- Planning JADs
- Design JADs.

Sometimes JAD workshops are called to concentrate solely on system requirements. These may be called Joint Application Requirements or JAR sessions.

4.3.2 Planning JADs

Planning JADs last approximately three weeks:

- One week preparation
- Two to three days session
- One week wrap-up.

Planning JADs can be used on a project for defining:

- Objectives
- High-level requirements
- Design session scoping

The number and purpose of planning JADs will depend upon the size of the project, how it is partitioned into deliverables, and the experience of you and the JAD leader in using and customizing JAD.

4.3.3 Design JADs

There are two types of design JAD:

- System definition or application definition design JAD, which is used to define requirements into successively lower levels until the system is fully defined.
- Design JAD, which is used to define the user's external view of the system (external design).

Design JADs last varying lengths of time, never less than three weeks:

- Preparation = same number days of session
- Session days
  - Application definition: half a day for each high-level process identified during the JAD plan
  - Application design: one day for each six to eight screens of medium complexity.
- Wrap-up = same number days of session.

A project may have a large number of design JADs for each planning JAD.

4.3.4 Advantages of JAD

A JAD:

- Brings key users together to discuss and make critical decisions for the project
- Eliminates the need for the design analysis to 'interpret' conflicting user input
- Fosters consensus on critical project decisions and user designs
- Draws upon executive management, IS, user management, end users, and the project team to gain commitment to the project
- Allows you to participate more actively since facilitation is handled by a JAD leader
- Gives the users more responsibility for the external design of the system.

4.3.5 Disadvantages of JAD

You might find the following problems with JAD:

- You must delegate critical contact with the user management to the JAD leader
- Scheduling users for extended periods of time can run into a number of conflicts (especially without strong executive sponsor support)
- Can sometimes be "design by committee"
- Can inhibit innovation. The "conservative" majority in a sizeable group of people may overrule the radical, lateral thinker
- In intensive sessions the concentration on the detail may cause of loss of sight of the original intention of the system. For example, an argument on whether a Post Code field should be 8 or 10 characters may miss the question of why an address is being entered at all. The JAD leader should be skilled in spotting these situations.

4.3.6 JAD Players and Roles

The JAD team should consist of:

- JAD leader
  - Responsible for the success of the JAD
  - Neutral, unbiased facilitator
  - Manages the entire JAD effort
  - Produces the JAD deliverable (as required by the executive manager, project administrator, and you)
- JAD analysts
  - Responsible for documenting the JAD results
  - Produces the JAD deliverable
  - Remains neutral and unbiased during JAD session and does not contribute to the content of the document

  Note: More than one analyst may be required, depending upon the complexity of the deliverable, for example for modelling.
- Executive sponsor
  - Commits to the JAD process

- Empowers and commits the JAD session participants
- Resolves issues not addressable during the JAD sessions
- Approves the JAD deliverable

- JAD Project Administrator (user Project Manager)
  - Acts on behalf of the executive sponsor during the JAD session
  - Responsible for change control
  - Assists the JAD leader and JAD analyst in preparing for session
  - Sets up interviews with critical participants
  - Ensures facilities and supplies for the JAD are available
  - Enforces the executive sponsor's commitment of resources
- User management
  - Responsible for the content of the JAD deliverable
  - Responsible for committing department resources for further project activities
  - Responsible for making critical project requirement decisions.
- End users
  - Responsible for critical design decisions
  - Responsible for the content of the JAD deliverable
- IS (client IS department)
  - Responsible for legacy systems integrity during JAD sessions
  - Plays an advisory role to user management and end users during JAD sessions regarding technical and support boundaries
  - Acts as advisor to the project team
- Project architect
  - Responsible for gathering enough information to assist in making appropriate system architecture decisions (should work with JAD leader to identify required data)
  - Plays an advisory role to user management and end users during JAD sessions regarding technical and support boundaries
  - Acts as advisor to the project team.
- Project Manager
  - Responsible for gathering enough information to assist in making appropriate project decisions (should work with JAD leader to identify required data)
  - Plays as advisory role to user management and end users during JAD sessions
  - Acts as advisor to the project team.
  - Assists the JAD leader with introductions to critical project participants, including the executive sponsor and the user project manager
  - Assists in the creation of the JAD deliverable
  - Assists the JAD leader with any personnel issues that might come up during the JAD process
  - Commits project team members to the JAD process.

4.3.7 JAD Tools Assistance

The JAD analysts may employ on-line tools to enhance the productivity of the JAD session. A quick prototyping tool may be of benefit if attendees can rapidly see prototype application screens as they are discussing designs and requirements. Indeed, in some typical JAD workshops the discussions and agreements are triggered by the display of a prototype of the application (via overhead projector for large audiences).

5 Organization

AD projects generally demand increased emphasis on the following areas within the Organization activity of MITP.

Subtopics
5.1 AD Subproject Manager
5.2 Key Personnel

5.1 AD Subproject Manager

Typically, the AD project has its own Project Manager reporting to the Project Manager of the business project. This person is referred to as the IS project leader in the following discussion.

5.2 Key Personnel

The key personnel in this project organization are:

IS project leader
For projects involving many people, especially if more than one department is involved, you might appoint an IS project leader to assist with leading the work and preparing the project control documentation. However, this does not mean that you can relinquish the formal project management responsibility.

The IS project leader is responsible to you for building the IS system. For small projects you could fill this position. The IS project leader also has a responsibility to the AD department, for example, to ensure that standards are followed, and that the system meets operational requirements, as well as to the project, and also uses services provided by the AD department, for example, a data administration group may be involved in data analysis and database design. For some projects there may be more than one IS subproject, each requiring an IS project leader.

Test leader
For projects involving many people, especially if more than one department is involved, you might appoint a test leader to assist with leading the testing side of the project.

The test leader leads the test teams and takes responsibility for testing the deliverables from the project. The test leader acts independently from the IS project leader and the development team.

User project leader
For projects involving many people, especially if more than one department is involved, you might appoint a user project leader to assist with leading the user side of the project.

The user project leader is responsible for approving the user requirements for a system and is supported in this by a requirements group that represents all parties affected by the system. The user project leader also takes responsibility for work within the user departments.

In many cases the above roles may overlap, and one or more may be filled by the same person. Also, depending on the size of a project, additional levels of management may be required.

Project Office
Within the Project Office, an additional function that may be required for a large project is a technical architect who oversees the design of the system being developed.

Other key positions to be filled can include:
Steering group members
The steering group ensures long-term requirements are met and that interfaces with other parts of the enterprise are handled effectively.

The group agrees and commits the overall scope, cost, timescale, and resources for the project and monitors progress and priorities, resolving any points of dispute and approving implementation plans.

Requirements group members
The requirements group is led by the user project leader, that is, the person within a user department who is responsible for representing the users to the project. The group represents all parties affected by the system, including the eventual owner of the developed system.

6 Project or Phase Startup

Following project definition, it is important to get the project work started as quickly as possible. To achieve this, the project organization must be established, staff appointed, facilities obtained and set-up, and the project management systems initiated. Some of these activities are also required at key milestones.

The size and formality of this process varies widely according to the project category. A small project requires very little effort to get started.

Subtopics

6.1 Configuration Management

All of the processes associated with configuration management require initiation by the IS project leader. These include version control, library control and promotion procedures for modules, regression testing, and release management.

6.2 Define Project Policies

The project definition report defines the overall project approach, but there are often some lower level policies that need agreeing at the outset of the project. These may include:

- Relationships with suppliers, for example, a software house
- Support of the system after implementation.

7 Project or Phase Planning or Re-planning

AD projects generally demand increased emphasis on the following areas within the Planning activity of MITP.

Subtopics
7.1 Identifying and defining Tasks
7.2 Detailed Quality Planning
7.3 Quality Testing

7.1 Identifying and defining Tasks

The actual AD method employed will provide a definition of most AD tasks.

7.2 Detailed Quality Planning

These list the deliverables of the project, how their production is controlled, and how their quality is assured. In an AD project, the quality plans should include testing plans and how the outputs are controlled. Inspection plans, defect tracking, and assurance reviews all need to be included.

The quality plan ensures all project deliverables are of the highest quality. It includes:

- Approval and change responsibilities
- Techniques, tools, and methods to be used to assure quality in deliverables.
- Quality checks to be performed.
- Target defect density for deliverables.
- Rework policy for deliverables failing quality checks.
- Testing plans. The process for producing these should be started in the external design stages. At this stage, test cases need to be planned in detail and signed off against the design. You are recommended to appoint a separate team (not part of the development team) and schedule enough time for the completion of testing, which can take up to 20% of the project. In the testing plans there should be processes implemented to allow the tracking of errors. Regression testing also needs to be scheduled as part of the test plan.

7.3 Quality Testing

Unless testing is adequately planned and executed, the quality of the product becomes unacceptable and may cause missed implementation dates and unhappy product users.

Testing should be an ongoing process throughout the development life cycle to ensure that what is being developed is what should be developed.

To achieve this, the approach and extent of testing should be planned and agreed at the same time as, and as part of, the project definition.

This testing should be based on the development deliverables.

It is important that developers and users understand the role they play, their level of involvement, and the anticipated testing methodology to be use in advance of the actual testing.

7.3.1 Planning

The test plan describes:

- Overall testing objectives
- Overall testing strategy, including testing stages, test types, integration strategy, interface testing, and user involvement

- Overall milestones and schedule
- Each testing stage, corresponding objectives, methods, roles and responsibilities, dependencies, entry and exit criteria, resource requirements, and schedule
- The test method for each key deliverable
- Any deviation from standard error, issue, and problem management processes
- Test Entry and Exit criteria
- Any tools used, for example, logging and tracking databases, simulators, automated testers (robots) etc.

7.3.2 Methods and Techniques

The following techniques can be used within a quality plan:

Checks

Desk checks, checks, and multichecks are all informal tests of deliverables that occur when the deliverable has been built but has not yet been handed over for more formal tests, such as an inspection. These checks should identify the more obvious errors and omissions, enabling correction before the more formal tests. These checks are usually very cost effective.

- Desk check: the deliverable is checked by the author by looking through it from start to finish, trying to spot any errors or omissions.
- Check: the deliverable is checked by one person other than the author by looking through it from start to finish, trying to spot any errors or omissions.
- Multicheck: the deliverable is checked by at least two people other than the author. Each person looks through the deliverable from start to finish, trying to spot any errors or omissions.

Inspections

Inspections provide a formal and rigorous method of performing technical reviews at the end of development phases.

AD inspections can be characterized as follows:

Inspections appear as separate scheduled activities in the project development plan, and the project schedule contains a time allowance for rework of deficiencies identified in the inspection process.

Each development phase (at the end of which work products are to be inspected) is defined, as are the exit criteria for each.

At the end of an inspection, formal approval is required that deficiencies have been satisfactorily reworked and exit criteria have been satisfied before work can proceed to the next development phase.

A specially trained moderator schedules and conducts the inspections and chooses the participants, who are usually selected on the basis of special skills or knowledge. The moderator is not the developer of the product being inspected nor a member of the development team. The moderator does not usually devote full time to this role, but is a working analyst or programmer. A moderator's work may be inspected in turn by inspectors from other projects.

Simulations

A simulation exercise is a document-format standard process for removing defects from programming activities.

It is basically a manual, form-based walkthrough of application function and logic. Normal participants are the developers, a moderator, and members of the test team.

Test Identification
How do you know what should be tested?

The tests that you need to be identified should cover:

- Normal processing of valid situations, for example, correct data handling
- Normal processing of error situations, for example, incorrect data handling
- Abnormal processing situations, for example, system corruption or a problem with the processing environment.

Test Documentation

Test Situation Matrixes
These provide a concise means of documenting the tests to be executed. Matrixes are used for documenting both program unit tests and other types of test.

Test Cases
Test cases are derived from the test situation matrix. Each test case should identify the start position, all the data and data values and the processing necessary to perform the situations that it is to test.

A test case should also identify the expected end position and all the expected data and data values resulting from performing the situation being tested.

Test Scripts
These fully identify a set of data and processing combinations which together describe, in a step-by-step manner, how to prepare and run a test to check whether the predicted results are being achieved from the module, program, or application being tested.

Each script should contain:

- A description of what is to be done
- Who will do it
- What to look for
- Cross-references of data to test situation and test cycle
- A description, in step-by-step format, of:
  - The actions performed on the base data
  - The results expected for specific situations.
- References for future regression testing.

Each test script has a specific scope and seen together they are designed to cover all the testing of the module, program, or application.

Test Data

Test data is coded directly from the data and data values recorded in the test cases. For each test case, the data coded is both the base start position data before the test is run and the expected results end position data should result after the test has run. Both sets of data are then stored ready for running the test for the test case.

Checking Results
Results checking uses a test run to compare actual and expected results. Mechanized comparison facilities greatly simplify the identification of the differences and are not likely to miss or overlook differences.

Differences can only be attributed to one of three things:

- The input data is wrong. It could have been wrongly interpreted or wrongly coded.
- The expected results are wrong. They could have been wrongly interpreted or wrongly coded.
- The module, program, or application is wrong. It has been developed such that it contains an error.

Therefore the input data, expected results, or the module, program, or application have to be corrected and the test rerun.

The test is finished successfully when no differences remain between the actual and expected results.

7.3.3 Test Types

The following is a list of test types that may be employed in an AD project:

Unit test

Unit testing is designed and carried out by the programmer or programming team. Its major objective is to make certain that the program executes as intended without abnormal ending. The test operates at individual module or program function level.

Unit testing ensures that:

- All paths of the code are exercised
- All paths end appropriately
- Conditions are tested to their extremes
- External interfaces are as expected
- Bad input data is handled appropriately
- The module or program meets its specification.

Function test

Function testing is designed and carried out by the test team. Its major objective is to make certain that the individual programs perform the business purpose without abnormal ending or giving invalid results.

The testing should cover:

- On-line and batch functions
- Compliance to standards
- Transaction and screen flow
- Fields, options, keys, exits, and return codes
- Data validation, data handling, and error handling
- Bridges, interfaces, and help facilities

Integration test

This proves that the programs function together as defined by the external design.

An integration test should test for interaction between programs or between programs and an external interface. It is in effect just the joining of two or more function tests where the functions flow together. This test is planned and performed by the test team.

Integration tests are performed as new tested functions are added to the system, until the complete system has been assembled. The last integration test is the full integration test (system test).

Business test

This proves that the application functions support the required business functions. The business test should be developed jointly by the test team and business representatives. It should test the key business algorithms, system input, and system output.

Audit and controls test

This proves that the audit and controls facilities provided by the system or application function correctly as described in the external design document. It should test the audit and control information maintained by the system or application and ensure that this information is correct and cannot be modified by bypassing or abusing the application or abusing system functions. This test is performed by the test team.

Security test
This proves that the system or application security cannot be bypassed or broken. Security requirements have been defined in the external design.

The security test should test system, application and data access, attempts by a user to make unauthorized changes (including using non-application functions), the classification of reports and data, and encryption of stored or transmitted data (where encryption is used). This test is performed by the test team.

Tests should confirm that security functions (such as RACF) are functioning correctly and that security definitions are correct (read only and write access to data is limited to appropriate users).

Production environment test
This ensures that the application functions correctly, as defined in the external design, in the production environment.

This test is particularly important when the development and production systems are significantly different. It should be co-ordinated with the implementation test and the package and unpackage test. This test is performed by the test team together with the appropriate computer operations and support groups.

Backup and recovery test
The application must be capable of being restarted from failure. This may involve restoring data, backing out changes, and restarting or cancelling jobs. This test is performed by the test team together with the appropriate computer operations and support groups.

Volume test
This ensures that the application supports the volumes of data and transactions described in the external design. The results of this test prove that the application performs in a completely satisfactory manner when presented with the expected business volumes.

It tests that, given various key volume parameters, for example, normal and peak volume of a transaction or total volume of a data record, the application meets stated criteria. This test is performed by the test team, possibly together with the computer operations group.

Performance test
This ensures that the application supports the performance criteria described in the external design. It tests that, given various key volume parameters, for example, normal and peak volume of a transaction, the application meets stated criteria. It shows that system performance meets stated performance criteria, for example with simulated transaction load "X", terminal response (stating how measured) is "Y" or better.

This test is performed by the test team, possibly together with the computer operations group. The performance test may be run together with, and certainly should be co-ordinated with, the volume test.

Stress test
This evaluates the effect of stress situations on application performance. It tests the response of the application to identified stress conditions, for example, peak transaction volumes or the peak limit numbers of concurrent users, and identifies actual and potential bottlenecks. This test is performed by the test team together with the computer operations group.

The situations that should be tested include:

- Contention for data causing similar to a traffic jam, stopping or severely restricting processing
- Known peak processing traffic or times
- Unusual but possible peak processing times
- Extra processing when recovering from a failure, causing peak loading.

Usability test
This ensures that the system or application is functionally usable by users, operations, and support staff. Its objective is to confirm that the design is usable rather than the product conforms to the design. The results of this test identifies whether the application, documentation, and other user interface components of the system or application are designed to aid the users rather that hinder them.

The usability test should cover all application functions and related user documentation, and involve representatives of all types of user. This test is performed by the users supported by the test team.

Full integration test (system test)
This proves that all programs function together as defined in the agreed external design. The purpose of the test is to verify the functions of the completely constructed system or application run as expected. This testing ensures that all functions work together and that data passed between them is correct.

This test is performed when all the individual component programs have been integrated into a single system or application, and is the final integration test. The test is performed by the test team.

Package and unpackage test
This ensures that the system can be "packaged" and the package installed in another system environment. It is particularly applicable where an application is developed on one machine and used on another or when an application is to be installed on a site that is remote from the development location. The results of this test prove that all components of the application can be loaded successfully on to a transmission medium, and that the application can then be unloaded, generated, and verified at another location.

The package and unpackage test should cover the packaging and unpackaging procedures to be performed at the development and target sites. This test is performed by the test team, by operations or by the end-users themselves (for example, for PC applications). The test should cover the package and unpackage procedures and any tools used, and should be co-ordinated with the implementation test.

Migration test

This proves that the migration process from the current application to the new application works as defined in the external design. It covers the migration of data, application and support. The process tested should exactly simulate the process to be followed when the developed product is migrated to production.

This test is performed by the test team, implementation team, operations, or a combination of these.

Implementation test

This proves that the implementation for the new system or application works as defined in the external design. It covers data loading, application implementation, and support implementation. The results of this test should prove that the system or application can be installed, set up, and run in production.

The implementation test performs a representative test implementation following the documented implementation procedures. This test is performed by the test team, implementation team, operations, end-users or a combination of these.

Regression test

Regression testing is carried out whenever changes are made, to ensure that everything works precisely as it did before the change. It ensures that the existing function of the system has not been affected by the change. This test is performed by the test team.

User acceptance test

This ensures that the users find the delivered system acceptable. It enables the user to simulate live business operations, including mechanized and manual procedures, forms handling, and processing schedules. The results of this test should prove that the system satisfies the original user requirements, matches the agreed business process design, is usable, can handle projected data volumes, and meets performs requirements. The system, documentation, and all user interfaces are tested to ensure they are complete and correct.

Operations acceptance test

This allows operations to gain experience, and hence acceptance, of the operational features and performance of the system. The test ensures that operations find the delivered system acceptable.

8 Tracking and Reviewing

AD projects generally demand increased emphasis on the following areas within the Tracking and Reviewing activities of MITP.

Subtopics
8.1 Tracking and Reviewing Procedures
8.2 Milestone Tracking
8.3 Work Package tracking

8.1 Tracking and Reviewing Procedures

The MITP principles of project tracking demand a minimum tracking system that considers achievements to time and costs to budget. These principles are discussed in detail in the MITP Progress Tracking Guide, which also provides sample and skeleton reporting forms.

The MITP principles of project reviewing relate to informing management of the status of the project and preparing for and taking rapid actions to correct out-of-line situations These principles are discussed in detail in the MITP Progress Reviewing Guide.

8.2 Milestone Tracking

As suggested earlier an AD project's milestones can best be determined from the AD method employed. The following observations are made:

The traditional waterfall development method (Requirements, Design, Build, Test and Implement) tends to provide enough granularity to be used to establish major project milestones which can be tracked

Rapid development techniques such as the three-phase Initiation Control Point, Continuation Control Point and Release Control Point method gives only three major milestones, which may not be granular enough for large project tracking. Judgement must be employed as to whether the AD method employed gives meaningful milestone tracking points without further breakdown.

Beware lack of visibility. One of the problems of AD projects, as mentioned earlier, is that software development tends to lack visible progress indicators. This is particularly true in the earlier stages of Requirements Gathering and Design. These stages normally occupy large segments of elapsed time within the overall schedule and can be "paper" exercises in as much as no actual programming may be taking place. Other than a pile of paper, the sponsor or other outside observer may not be able to detect material progress within the project. Prototyping and iterative development techniques reduce the problem, but you should consider breaking down these phases into smaller, more demonstrable, segments.

8.3 Work Package tracking

The common problem within AD projects is the "90% complete" syndrome. Programmers can provide erroneous tracking information by claiming that a work effort (for example, a module) is nearly ready whenever status is requested. Work Package tracking, where the breakdown is taken to application module level (for example) helps to avoid this. Always track task starts AND finishes.

9 Managing Exceptions

AD projects generally demand increased emphasis on the following areas within the MITP Manage Exceptions technique.

The MITP technique of Exception Management covers Problem Management, Issue Management, Change Management, and Error and Fault Management. See the Exceptions Management Guide. For AD projects, the important activities are Change Management and Error and Fault Management.

Subtopics

9.1 Change Management

9.2 Error and Fault Management

9.1 Change Management

For an AD project, you must deal with change management differently. It is normal to make an allowance for change, which is usually estimated between 5% and 10% of the development effort, but could vary considerably depending on the nature of the project. The allowance should be managed against signed off changes through a rigorous change control process.

It is not possible to accomplish all of the changes towards the end of a project and thus the allowance should be adjusted proportionately downwards depending on the stage of the project. For example, if the project is half way through and no change allowance has been used, up to 50% of the change allowance should be cancelled.

MITP defines a change management system to control the effect of changes on the project. In an AD project the system has the following elements:

- A potential change is documented and logged on a change log.
- The change is analysed for its benefit and effect on project timescales and costs, and on any change allowance outstanding.
- The change is accepted, deferred (to the next release of the system, for example) or rejected, after consultation between the defined parties. If the change is necessary and cannot be accommodated within the change allowance, an escalation process should be implemented to review what actions are required
- Accepted changes are incorporated into project plans
- Project reviews regularly monitor the change log.

Changes are inevitable in every project, and so it is essential that you set up a system to allow you to manage the changes. The key aspect of managing changes is communication:

- Everyone who needs to know about any change should be told that it is about to happen, so that they can comment as appropriate and their comments can be considered in deciding whether or not to accept the change
- The result of the decision should be communicated to the project so that the plans of the various parts can be kept in step.

The following are the essentials of such a system, irrespective of the size of the project:

- All requests for change should be directed to one central point - a list or log
- All change requests should be filtered by a responsible person representing the sponsor or user community, as agreed at project definition time. Only allow "approved" change requests to enter the review process.
- The consequences of implementing each change in each of the affected areas must be assessed and this information fed into the decision making process
- You initial project planning and estimating should include a significant allocation of resource effort for change request analysing and reviewing. This can be a major task, and it is important to get it right. A one-line change request may significantly impact system design and implementation. This has to be correctly analysed and the potential impact accurately sized. This takes time and effort.

Although this relates to change requests, some changes are imposed rather than requested, for example, by executive management, and some are enforced by circumstances, for example, change of responsibilities due to absence through illness.

But they should all go through the same process to ensure that they are recognized, communicated, and assessed.

9.2 Error and Fault Management

An error is a fault that has been introduced during the development life cycle.

There are two types of error within an AD project:

1. Errors discovered within the development phase in which they were originally caused
2. Errors discovered in a later development phase from that in which they were originally caused.

Errors discovered within their development phase can normally be corrected within the effort allowance and timeframe of the phase. This is very much business-as-usual.

Errors discovered in a later development phase can cost significantly more effort and time to correct. The correction of such errors is treated as correction rework.

The amount of correction rework allowance for the project needs to be considered and defined at project definition time and to be planned in at project planning time. If the 'correction rework' cannot be accommodated within the 'correction rework' allowance, the project contingency allowance takes up the excess correction rework cost.

If the project contingency allowance is insufficient to cover this excess correction rework cost, a project change request needs to be raised to request a change to the project plans and costs.

The following is a suggested process for managing errors discovered in later development phases:

- Set up an error management system for errors requiring correction rework, where:
  - The error can be logged, and fully described
  - Correction rework plans, progress and effort can be recorded
- Estimate the probable effort for rework, considering:
  - The number that can arise, and their size
- Put in plan an effort allocation for correction rework, considering:
  - It as a project overhead
  - The maximum acceptable effort
  - It tends to occur earlier rather than later in the project.
- Track and report on correction rework
- Track and report on remaining effort allocation for correction rework.

10 Project or Phase Completion

The purpose of the MITP project or phase completion process is to mark and record the formal ending of a project or phase, and in the case of a project completion, to put in place a process for management and control of the achievement of business benefits. It should also ensure the recording of information about the management of the project which could be of value to other project managers, or which may indicate a need to alter the project management standards. In an AD project it is vital to capture information on development to allow the building of information that can be used for future projects.

Subtopics
10.1 Project or Phase Completion Documentation
10.2 Plan Post-Implementation Review

10.1 Project or Phase Completion Documentation

Facts about the project are collected and archived. These include:

- Effort and timescale spent on tasks and phases
- Size and constitution of project team
- Product size information, for example:
  - Numbers of programs, screens, reports
  - Function point count
  - Pages in document deliverables.

10.2 Plan Post-Implementation Review

The final process in a MITP for AD projects project is to plan a review of the newly implemented system. The review should normally be held between three and four months after implementation and concentrates on:

- An evaluation of the benefits accruing from the new system
- Problems with the new system
- Future releases and their content.

11 Goals and Benefits

The recommended method of assessing goals and benefits is to hold a Post-Implementation Review after the completion of the development project.

A Post-Implementation Review is normally planned to take place when a developed system has been in live production use for a few months. This time should be sufficient to enable production running experiences and figures to have been gathered. This is often between three and four months after cutover.

These experiences and figures are matched against those predicted during the original development time, any conclusions or necessary actions are identified and appropriate plans are made to improve the system or the development processes, methods, or techniques.

Subtopics

11.1 Preparing for the Review

The preparation process involves:

- Gathering performance predictions.
  Ensure that the performance predictions from the project development stages of the development process are made ready and available for the review by confirming the figures are recorded, contain all the required predicted figures, and are available to the Post-Implementation Review.
- Recording service level agreements versus actuals for:
  - Run schedule
  - Availability
  - Volumes
  - Timings
  - Peaks
  - Planned maintenance effort
  - Unplanned maintenance effort.
- Gathering actual performance figures for:
  - Errors and abnormal ends
  - Operability
  - Control and security
  - Auditability.
- Agreeing attendees to the review:
  - Owner
  - Developers
  - Maintainers
  - Operations
  - Users
  - Technical support.
- Agreeing review content:
  - Management summary - history to-date
  - Present production key numbers
  - Benefit assessment against business case
  - User evaluation
  - IS evaluation
  - System control evaluation
  - Operability evaluation
  - Auditability evaluation
  - Conformation to agreed design
  - Conformation to standards
  - Conformation to privacy rules
  - Change request situation
  - Immediate concerns
  - Longer term concerns
  - Overall satisfaction rating
  - Planning for further user satisfaction reviews.

To confirm the content of the Post-Implementation Review itself, basing the list of topics on that originally recorded in the project definition report.

11.2 Running the Review

When running the review it is important to:

- Involve all agreed attendees to ensure that all the various functions attend the review and actively participate in it. If all agreed functions cannot attend, then the review should be rescheduled. Deputies may be permitted if they have the appropriate status and have knowledge of how the system has been since cutover to production.
- Cover all agreed content using the review material where appropriate and noting the review finding as the review progresses.

11.3 Reporting the Review Results

Normally report the review results, ensuring that you:

- Cover all agreed content
- Do not water down the results.

Record the findings of the Post-Implementation Review as a formal report, writing up the findings of the review from the notes made at the time and highlighting any action plans that were suggested.

The action plans resulting from the review may be processed as follows:

1. Note any necessary improvement (change) requests resulting from the Post-Implementation Review and to make them input to the regular development life cycle.
2. Plan to fix faults found in the system once in production and identified or raised at the Post-Implementation Review. These are those faults which should be fixed as ongoing maintenance (not handled as a development project). Faults to be fixed as system enhancements, following the development life cycle starting at the regular planning stage, are excluded.
3. Plan for the lessons learned from the actual system performance to be reflected into any planning guidelines or any relevant local standards and procedures by buildings a list of the significant lessons learned from the systems actual performance and passing the list to those responsible for maintaining the local standards, procedures and planning guidelines.
4. Obtain agreement to the actions planned following the Post-Implementation Review from those functions that would be affected by the actions by reviewing the actions planned with the affected functions and obtaining their consent to the actions.

12 MITP for AD Projects Checklist

The following standards for projects for the AD projects are met?

Question	Y/N
Does the project require a sponsor?	
Does the project have a manager?	
Does the project have a contract?	
Is the project sufficiently documented to define clearly:	
Objectives, scope and key deliverables	
Project category	
Acceptance criteria	
Organization and structure	
Roles and responsibilities	
Assumptions and risks	
Resources and costs	
AD Standards, processes and procedures to be used	
Does the project have clear start and end points?	
Is all the work planned, estimated and scheduled?	
Is task progress, resources, and costs tracked and managed?	
Are risks, issues, problems, errors and faults, changes and actions tracked and managed?	
Is the project status reviewed and reported regularly?	
Are plans kept up-to-date to reflect project state and direction?	
Are all key deliverables tested for quality?	
Project prerequisites	
Project sponsor identified?	
Project manager identified?	
IS project leader identified?	
User project leader identified?	
Test leader identified?	
Feasibility confirmed?	
Business case confirmed?	
Project definition:	
Goals and objectives	
Scope	
Key deliverables definition	
Has a JAD been held?	
Acceptance criteria	
Project structure	
Project milestones	
Assumptions	
Organization, roles and responsibilities	
Overall project schedule	
Overall project costs and resources	
Process management	
Project management	
Project office	

Quality management	
Change management	
Action management	
Version management	
Issue and problem management	
Error and fault management	
Risk management	
Progress management	
Contingency management	
Project assurance	
Development process	
Test process	
Acceptance process	
Implementation process	
Maintenance process	
Improvement process	
Project metrics	
Technical environment	
Facilities	
Service levels	
Quality plan	
Define risks	
Project organization and startup:	
Establish project organization	
Appoint staff	
Establish facilities	
Planning and replanning:	
Planning	
Estimating	
Scheduling	
Progress tracking:	
Task progress	
Resources	
Costs	
Reviewing	
Managing exceptions:	
Risks	
Issues and problems	
Errors and faults	
Changes	
Actions	
Progress (tasks, resources and costs)	
Project completion	
Plan for Post-Implementation Review	
Goals and benefits:	
Post-implementation review	

Index

Readers Comments

MITP
Application Development Projects Guide
Version C5.0

Publication No. MICG1ADP

Overall, how satisfied are you with the information in this book?

Legend:

1	Very satisfied
2	Satisfied
3	Neutral
4	Dissatisfied
5	Very dissatisfied

	1	2	3	4	5
Overall satisfaction					

How satisfied are you about the information this book contains:

	1	2	3	4	5
Accurate					
Complete					
Easy to find					
Easy to understand					
Well organized					
Applicable to your task					

Please tell us how we can improve this book:

Allturn Group International B.V.
Groenendal 7c
5405 AS Uden (NB) The Netherlands
Email: Info@AllturnGroup.com

Phone: 0031 (0) 6 20 35 67 51

Name . . . . ______________________________
Company or Organization ______________________________
Address . . . . ______________________________
____
____
Phone No. . . . . ______________________________