Microsoft Solutions Framework

White Paper

Published: June 2002 For more information on MSF, see: http://www.microsoft.com/msf

MSF Team Model v. 3.1

Contents

Credits

Paul Haynes, Program Manager, Microsoft Solutions Framework

Allison Robin, Director, Microsoft Solutions Framework

Enzo Paschino, Program Manager, Microsoft Solutions Framework

Mark Short, Program Manager, Microsoft Solutions Framework

Pervez Kazmi, Program Manager, Microsoft Solutions Framework

Rob Oikawa, Program Manager, Microsoft Solutions Framework

Scott Getchell, Program Manager, Microsoft Solutions Framework

Laura Hargrave, Technical Editor, Microsoft Solutions Framework

Mike Lubrecht, Technical Writer, Microsoft Solutions Framework

Nany Huber, Technical Editor, Microsoft Solutions Framework

Suzana Vukcevic, Program Manager, Microsoft Belgium

Paulo Henrique Leocadio, Microsoft Consulting Services, Brazil

Paulo Rocha, Microsoft Consulting Services, New Zealand

Dolph Santello, Microsoft Consulting Services, US

Ralph Schimpl, Director, Microsoft Austria

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Part Number: 602-i400a

Abstract

The Microsoft Solutions Framework (MSF) Team Model describes Microsoft's approach to structuring people and their activities to enable project success. The model defines role clusters, functional areas, responsibilities, and guidance for team members to address so that they can reach their unique goals in the project lifecycle.

Introduction

To maximize the success of IT projects and operations throughout the entire IT life cycle, Microsoft Solutions Framework and Microsoft Operations Framework (MOF) provide guidance and proven practices for effectively planning, building, deploying, and operating solutions. This information is derived from the experience gained within Microsoft on large-scale software development and service operation projects, the experience of Microsoft's consultants, and common best practices from the worldwide IT industry. It is delivered in the form of white papers, guides, tools, templates, case studies, and courseware. The guidance and practices are organized into two complementary and well-integrated bodies of knowledge.

Microsoft Solutions Framework

Creating a business solution on time and within budget is simplified by using a proven approach. MSF provides proven practices for planning, building, and deploying successful IT solutions. As opposed to a prescriptive methodology, MSF provides a flexible and scalable framework to meet the needs of any size organization or project team. MSF guidance consists of principles, models, and disciplines for managing the people, process, and technology elements that most projects encounter.

For more information on MSF, see: http://www.microsoft.com/msf

Microsoft Operations Framework

MOF provides guidance that helps enable organizations to achieve mission-critical system reliability, availability, supportability, and manageability of IT solutions built using Microsoft products and technologies. The principles, models, and disciplines of MOF address the people, process, and technology issues pertaining to operating complex, distributed, heterogeneous IT environments.

For more information on MOF, see: http://www.microsoft.com/mof

For more information on the IT Infrastructure Library (ITIL), the industry best practices on which MOF is based, see: http://www.itil.co.uk/index.html

Team Model Fundamentals

The MSF team model was developed over a period of several years to compensate for some of the disadvantages imposed by the top-down, hierarchical structure of traditional project teams.

Teams organized under the MSF team model are small and multidisciplinary, in which the members share responsibilities and balance each other's competencies to keenly focus on the project at hand. They share a common project vision, a focus on deploying the project, high standards for quality and communication, and a willingness to learn. This paper describes the various role clusters within the team, along with their goals and functional areas. Guidance is also provided on using the Microsoft approach to teaming when scaling for both small or large and complex projects.

The foundation principles, key concepts, and proven practices of MSF as they apply to the Microsoft team model are outlined below. The primary ideals are highlighted in this section and referenced throughout this document as the details of the MSF team model are discussed.

Underlying MSF Foundation Principles

MSF includes several foundational principles, cornerstones of the framework's approach. Some of the principles relating to working as a successful team are highlighted in this section.

Clear Accountability, Shared Responsibility

MSF combines a shared responsibility for doing work with a clear accountability for ensuring it gets done.

The MSF team model is based on the premise that each role has equal goals, presents a unique perspective on the project, and that no single individual can successfully represent all of the different quality goals. To resolve this dilemma, the team of peers needs to combine a clear line of accountability to the stakeholders with shared responsibility for overall success.

Within the team, each role is accountable to the team itself (and to their own respective organizations) for achieving their role's quality goal. In this sense, each role is accountable for a share of the quality of the eventual solution. Responsibility is shared across the team of peers (allocated in line with the team roles). It is interdependent for two reasons: first, out of necessity, since it is impossible to isolate each role's work; second, by preference, since the team is more effective if each role is aware of the full picture. This mutual dependency encourages all team members to comment and contribute outside their direct area of accountability, ensuring that the full range of the team's knowledge, competencies, and experience can be brought to bear. All team members own the success of the project; they share in the kudos and rewards of a successful project and are expected to improve their expertise by contributing to and learning from the lessons of a less successful one.

Empower Team Members

In an effective team, each member is empowered to deliver on their own commitments and has confidence that, where they depend on the commitments of other team members, that these will also be met. Likewise, the customer has a right to assume that the team will meet its commitments and will plan on this basis. At worst, the customer should be notified as soon as possible of any delay or change.

An MSF team provides members with the degree of empowerment they need to meet their commitments. In return, it relies on the integrity and motivation of all team members to:

- Be prepared to make commitments to others.
- Clearly define the commitments they undertake.
- Make every reasonable effort to deliver against those commitments.
- Communicate honestly as soon as they realize that a commitment may be at risk.

As soon as more than one person is needed for an activity, each participant’s efforts will be influenced by their dependencies on what other team members are doing. However they can’t spend time monitoring every dependency on which their own work may rely. Effective teams develop confidence that their colleagues are empowered and committed to the team’s objectives.

Consider the analogy of an athletic relay team. When the runner for the second leg starts running, the runner doesn’t slow down and look backwards to see how close the fore-runner is. Instead, the runner concentrates on accelerating as fast as possible and then simply stretches back to receive the baton, confident that it will be delivered. This confidence is based on practice, experience, and trust.

In a complex project, team members need to develop a similar level of trust and this trust is built every time a commitment, however small, is met. A few simple guidelines for engendering trust are:

- Empower team members to meet the commitments assigned to them. Empowerment requires that team members are given the resources necessary to perform their work, are responsible for the decisions that affect their work, and understand the limits to their authority and the escalation paths available to handle issues that transcend these limits.
- Be prepared to make commitments to others. Preparation includes state of mind (approaching meetings with a willingness to take actions), readiness, and understanding the implications of a commitment and its impact on current workload and resources. The corollary to this is to defer making major commitments until their implications are understood. Instead, team members should propose a smaller commitment that they do understand, such as to research the implications and come back shortly with a firm commitment. Successful delivery on the smaller commitment will build team trust.
- Clearly define the commitments that are undertaken. This avoids misunderstandings that can damage the confidence team members have in one another.
- Make every reasonable effort to deliver against those commitments. If a team includes people from different organizations, expectations of what is reasonable may differ. For example, some team members may assume it’s reasonable to work on weekends; others may see this as exceptional or may lack access to buildings on weekends.
- Communicate honestly as soon as a commitment may be at risk. Inevitably there will be times when things change, either due to some reprioritization, an unexpected event, or simply that a task took longer than expected. Early communication enables other dependent team members the opportunity to plan accordingly. Perhaps they can even suggest an approach that solves the problem.

In most organizations these behaviors are embedded in the culture and regarded as so clear that they are rarely discussed. However, MSF teams will occasionally need to work with organizations where these values are not fully understood and respected. These organizations often exhibit a high-blame culture that restricts an open flow of information. In these cases, the team leaders should clearly state their expectations in this regard and help new team members to adopt this way of working.

Focus on Business Value

The MSF team model advocates basing team decisions on a sound understanding of the customer's business and on active customer participation in project delivery. The product management role acts as the customer advocate to the team and is often undertaken by a member of the customer organization. Product management owns the business case, which provides continuity from earlier strategic work. Part of product management's responsibility is to ensure that key project decisions are based on a sound business understanding.

The release management role is explicitly responsible for ensuring smooth deployment and operations. In doing so, this role acts as a bridge between solutions development, solutions deployment, and on-going operations, ensuring that the project delivery group is continually aware of the impact its decisions might have on value delivery during production operations.

Shared Project Vision

MSF strongly advocates the adoption of a shared vision to focus the approach of a team, either towards delivery of an IT solution or towards provision of an IT service in an operating environment.

It is important to have a clear understanding of what the goals and objectives are for the project or process. This is because the team members and customers make assumptions on what the solution is going to do for the organization. A shared vision brings those assumptions to light and ensures that all participants are working to accomplish the same goal. The shared vision is one of the foundations of the MSF team model.

When all participants understand and are working towards a shared vision, they are empowered by the ability to align their own decisions to the broader team purpose represented by that vision.

Without a shared vision, team members may have competing views of the goal, making it much more difficult to deliver as a cohesive group. And if the team does deliver, members will have difficulty determining their success because it depends on which vision they measure it by.

Stay Agile, Expect Change

MSF assumes that things are continually changing and that it is impossible to isolate an IT solution delivery project from these changes. The MSF Team Model ensures that all core roles are available throughout a project so that they can contribute to decisions arising from these changes. As new challenges arise, the MSF Team Model fosters agility to address these issues. The contribution of all team roles to decision-making ensures that matters can be explored and reviewed from all critical perspectives.

Foster Open Communications

Historically, many organizations and projects have operated purely on a need-to-know basis, which frequently leads to misunderstandings and impairs the ability of a team to deliver a successful solution.

MSF proposes an open and honest approach to communications, both within the team and with key stakeholders. A free-flow of information not only reduces the chances of misunderstandings and wasted effort, but also ensures that all team members can contribute to reducing uncertainties surrounding the project.

The team of peers approach involves all roles in key decisions. It is one reason why the shared team vision is regarded as the essential start to the solution delivery process. It is also a foundation to the MSF risk management approach, which strongly advocates the involvement of all team members in risk identification and analysis and promotes a no-blame culture to encourage this. Open, honest discussion about what is working well and what can be improved provides the basis for the learning environment that MSF seeks to create.

There are a few important factors that may constrain the openness of the team’s communications, such as confidentiality of personal or commercial information. However, team members should question themselves whenever they decide to withhold information to ensure that the reasons for secrecy really are paramount. If they have built a relationship of trust through open communication, then on the rare occasions where they need to withhold information, they should be able to explain to their colleagues that there are over-riding reasons and ask for trust that these reasons are in the best interests of the project.

Key Concepts

Successful implementations of the MSF team model share several characteristics. These characteristics have been captured and are presented as key concepts in this section.

Team of Peers

The “team of peers” concept places equal value on each role. This enables unrestricted communication between the roles, increases team accountability, and reinforces the concept that each of the six quality goals are equally important and must be achieved. To be successful with the team of peers, all roles must have ownership of the product’s quality, must act as customer advocates, and must understand the business problem they are trying to solve.

Although each role has an equal value on the team, the team of peers exists *between* roles and should not be confused with consensus-driven decision making. Each role requires some form of internal organizational hierarchy for the purposes of distributing work and managing resources. Team leads for each role are responsible for managing, guiding, and coordinating the team while team members focus on meeting their individual goals.

Customer-Focused Mindset

Satisfied customers are priority number one for any great team. A customer focus throughout development includes a commitment from the team to understand and solve the customer’s business problem. One way to measure the success of a customer focused mindset is to be able to trace each feature in the design back to a customer or user requirement. Also, a key way to achieve customer satisfaction is to have the customer actively participate in the design and offer feedback throughout the development process. This allows both the team and customer to better align their expectations and needs.

Product Mindset

The product mindset is not about whether you ship commercial software products like Microsoft or develop applications for internal customers. It is about treating the results of your labor as a product.

The first step to achieving a product mindset is to look at the work that you are doing as either a project by itself or contributing to a larger project. In fact, MSF advocates the creation of project identities so that team members see themselves less as individuals and more as members of a project team. One technique Microsoft uses to accomplish this is to give projects code names. This helps to clearly identify the project, clearly identify the team, raise the sense of accountability, and serve as a mechanism for increasing team morale. Printing the team project code name on T-shirts, coffee mugs, and other group gift items are ways to create and reinforce team identity and spirit. This is particularly useful on projects with “virtual teams,” comprising elements from several different groups within an organization.

Once you understand that you work on a project, it’s just a matter of understanding that whatever the final deliverable is, it should be considered a product. Principles and techniques that apply to creating products, like those advocated in MSF, can be used to help ensure your project’s successful delivery.

Having a product mindset also means being more focused on execution and what is being delivered at the end of the project and less focused on the process of getting there. That doesn’t mean process is bad or unimportant, just that it should be used to accomplish the end goal and not just for the sake of using process. With the adoption of the product mindset, everyone on the team should feel responsible for the delivery of that product.

Former Microsoft program manager Chris Peters describes the product mindset as applied to software development in the following excerpt from a 1991 presentation:

“Everybody … has exactly the same job. They have exactly the same job description. And that is to ship products. Your job is not to write code. Your job is not to test. Your job is not to write specs. Your job is to ship products. That’s what a product development group does.

“Your role as a developer or as a tester is secondary. I’m not saying it’s unimportant—it’s clearly not unimportant—but it’s secondary to your real job, which is to ship a product.

“When you wake up in the morning and you come in to work, you say, ‘What is the focus—are we trying to ship or are we trying to write code?’ The answer is, we are trying to ship. You’re not trying to write code, you’re trying not to write code.”

Zero-Defect Mindset

In a successful team, every member feels responsible for the quality of the product. Responsibility for quality cannot be delegated from one team member to another team member or function. Similarly, every team member must be a customer advocate, considering the eventual usability of the product throughout its development cycle.

Zero-defect mindset is a commitment to quality. It means that the team goal is to perform their work at the highest quality possible, so that if they have to deliver tomorrow, they can deliver something. It's the idea of having a nearly shippable product every day. It does not mean delivering code with no defects; it means that the product meets or exceeds the quality bar that was set by the project sponsor and accepted by the team during envisioning.

The analogy that best describes this concept is that of the automobile assembly line. Traditionally, workers put cars together from individual parts and were responsible for their own quality. When the car rolled off the line, an inspector checked it to see if its quality was high enough to sell. But the end of the process is an expensive time to find all of the problems because corrections are very costly at this point. Also, since the quality was not very predictable, the amount of time required at the end to determine if it was sellable was not predictable either.

More recently in car manufacturing, quality has become "job one." That means that as work is being done (such as attaching a door or installing a radio), an inspector checks the work in progress to make sure that it meets the quality standards that are defined for that particular car. As long as this level of quality continues throughout the assembly process, then much less time and fewer resources are required at the end to ensure that the car is of acceptable quality. This makes the process much more predictable because the inspector needs to check only the integration of the parts, and not the individual work.

Willingness to Learn

Willingness to learn includes a commitment to ongoing self improvement through the gathering and sharing of knowledge. It allows team members to benefit from the lessons learned by making mistakes, as well as to repeat success by implementing proven practices of others. Conducting milestone reviews and blameless postmortems are components of the MSF process model which help teams commit to communicating. Teams that commit time in the schedule for learning, reviews, and postmortems create an environment of ongoing improvement and continuing success. In addition, one of the ways Microsoft is successful in creating a culture that is willing to learn is adding learning and knowledge sharing as part of individual review goals.

Motivated Teams Are Effective

Teams with low motivation suffer in two ways: Individually, the team members under-perform, leading to low quality and quantity of output; they also tend to work to narrow goals, and fail to appreciate the impact that their work has on colleagues. Both of these effects have a significant impact on IT projects, based as they are on a high degree of intellectual input and interaction.

MSF advocates devoting effort to building team morale and motivation. People who have worked at Microsoft recognize this as one of the company’s defining characteristics. Techniques that can be used to build motivation are:

- Clarify team vision.
- Build team identity, using project code-names and team paraphernalia—mascots, t-shirts, beakers, and so on.
- Spend time getting to know colleagues by way of social or team events.
- Schedule team-building sessions where team members can experiment with different ways of collaborating and interacting, normally outside the work setting.
- Ensure that the individual’s personal goals are considered, such as providing opportunities for personal or technical competency development, or managing the impact on work-life balance.
- Maximize the empowerment felt by individuals and listening to their views.
- Celebrate success.

Proven Practices

The following proven practices are common actions to members of an MSF team to ensure an ongoing focus for success.

Small, Multidisciplinary Teams

Small, multidisciplinary teams have inherent advantages, including the ability to respond more quickly than larger teams. Therefore, for large project teams it is better to create a team of teams—with smaller groups working in parallel. Team members with expertise or focus in specific areas are empowered with control to act where necessary.

Within teams or even within a role cluster, there are multiple disciplines that need a specific set of skills. People from various backgrounds, training, and specialization that comprise teams or roles all add to the overall product quality due to the unique perspective each brings to their role and ultimately the entire solution.

Working Together

One of the goals of the team model is to lower communications overhead so that teams have fewer obstacles to effective communication. Besides team structure, the geographic distribution and location of the team plays a major role in how effective a team can be with its internal and external communication.

In their study of Microsoft, *Microsoft Secrets*, Michael A. Cusumano and Richard W. Selby state:

"… Single-site development allows project personnel to get together physically on a regular basis and explore ideas interactively. Frequent and easy communication can prevent major problems from getting worse."

Having teams work together at a single site also helps to enforce the sense of team identity and unity.

Co-location such as working in the same section of a building, sharing offices, or setting aside space specifically for teams to gather has in the past proven to be the most effective method to promote open communication, which is an essential ingredient to the MSF team formula for success.

Although co-location is still the primary choice, the nature of business and the technological enhancements to communication available today do not prevent successful "virtual" teaming.

Virtual teams are teams of employees communicating and collaborating with each other primarily by electronic means. The communication occurs across organizational boundaries, space, and time. Collaborating in real time with colleagues through the Internet is profoundly changing the way people work and share information. The Internet is becoming a new standard of communication among team members, and collaborative software is paving the way for further productivity gains.

The notion of a virtual team is key, because without the organizational boundaries that encapsulate the roles into a coordinated unit, the virtual aspect requires even stronger communication, trust agreements, and relationships, explicit action plans, and automation tools that support tracking of projects and tasks so that action items do not get lost.

A vital component of a virtual team is the ability for each role to depend on and trust in the other roles to fulfill their responsibilities. This develops through a blend of culture, good management and, when possible, time spent working together at the same site.

Industry research finds that often little attention is given to communication skills or team fit when members are chosen for virtual teams. Analysts say this oversight is a key factor in the failure of many of these teams. When setting up a virtual team, look for members with the following characteristics:

- Can work independently.
- Demonstrate leadership skills.
- Possess specific skills required for the solution.
- Can share knowledge with the organization.
- Can help develop effective methods of working.

Total Participation in Design

Each role participates in creating the product specification because each role has a unique perspective of the design and its relationship to their individual objectives, as well as the team's objectives. This fosters a climate in which the best ideas from the various team perspectives can come to the surface.[1]

Team Model Overview

MSF is based on the belief that the six key quality goals must be achieved in order for a project to be considered successful. These goals drive the team and define the team model. While it is true that the entire team is responsible for the project's success, the team model associates the six quality goals with separate role clusters to ensure accountability and focus.

The six role clusters of the team model—product management, program management, development, test, user experience, and release management define common ways to identify a combined set of functional areas and their associated responsibilities. Many times role clusters are simply referred to as roles. Either way, the concept is the same: the framework and the team model are scalable to meet the needs of a particular solution. A role, or a cluster, may be one or many people depending on the size and complexity of a project, as well as the skills required to fulfill the responsibilities of the functional areas.

The MSF team model emphasizes the importance of aligning role clusters to business needs. Clustering associated functional areas and responsibilities, each of which requires a different discipline and focus, provides motivation for a well balanced team whose skills and perspective represent all of the fundamental project goals. Owning a clearly defined goal increases understanding of responsibilities and encourages ownership by the project team, which ultimately results in a better product. Since each goal is critical to the success of a project, the roles that represent these goals are seen as peers with equal say in decisions.

Note that these role clusters do not imply or suggest any kind of organization chart or set of job titles, because these will vary widely by organization and team. Most often, the roles will be distributed among different groups within the IT organization and sometimes with the business user community, as well as with external consultants and partners. The key is to have a clear determination of the individuals on the team that are fulfilling a specific role cluster and its associated functions, responsibilities, and contributions towards the goal.

Role Cluster	Goal	Functional Areas	Responsibilities
Product Management	Satisfied customers	Marketing Business Value Customer Advocate Product Planning	Acts as customer advocate Drives shared project vision/scope Manages customer requirements definition Develops and maintains business case Manages customer expectations Drives features vs. schedule vs. resources tradeoff decisions Manages marketing, evangelizing and public relations Develops, maintains, and executes the communications plan
Program Management	Delivering the solution within project constraints	Project Management Solution Architecture Process Assurance Administrative Services	Drives development process to ship product on time Manages product specification—primary project architect Facilitates communication and negotiation within the team Maintains the project schedule and reports project status Drives implementation of critical trade-off decisions Develops, maintains, and executes the project master plan and schedule Drives and manages risk assessment and risk management
Development	Build to specification	Technology Consulting Implementation Architecture and Design Application Development Infrastructure Development	Specifies the features of physical design Estimates time and effort to complete each feature Builds or supervises building of features Prepares product for deployment Provides technology subject matter expertise to the team
Test	Approve for release only after all product quality issues are identified and addressed	Test Planning Test Engineering Test Reporting	Ensures all issues are known Develops testing strategy and plans Conducts testing
User Experience	Enhanced user effectiveness	Technical Communications Training Usability Graphic Design Internationalization Accessibility	Acts as user advocate on team Manages user requirements definition Designs and develops performance support systems Drives usability and user performance enhancement trade-off decisions Provides specifications for help features and files Develops and provides user training
Release Management	Smooth deployment and ongoing operations	Infrastructure Support Operations Commercial Release Mgmt.	Act as advocate for operations, support and delivery channels Manage procurement Manage product deployment Drive manageability and supportability trade-off decisions Manages operations, support, and delivery channel relationship Provide logistical support to the project team

Satisfied Customers

Projects must meet the needs of customers and users in order to be successful. It is possible to meet budget and time goals but still be unsuccessful if customer needs have not been met.

Delivering the Solution within Project Constraints

A key goal for all teams is to deliver within project constraints. The fundamental constraints of any project include those of budget and schedule. Most projects measure success using “on time, on budget” metrics.

Build to Specification

The product specification describes in detail the deliverables to be provided by the team to the customer. It is important for the team to deliver in accordance with the specification as accurately as possible because it represents an agreement between the team and the customer as to what will be built.

Approve for Release Only after all Product Quality Issues Are Identified and Addressed

All software is delivered with defects. A key goal is to ensure those defects are identified and addressed prior to releasing the product. Addressing can involve everything from fixing the defect in question to documenting work-around solutions. Delivering a known defect that has been addressed along with a work-around solution is preferable to delivering a product containing unidentified defects that may surprise the team and customer later.

Enhanced User Effectiveness

In order for a product to be successful, it must enhance the way that users work and perform. Delivering a product that is rich in features and content but is not usable by its designated user is considered a failure.

Smooth Deployment and Ongoing Operations

Sometimes the need for a smooth deployment is overlooked. The perception of a deployment is carried over to the product itself, rightly or wrongly. For example, a faulty installation program may lead users to assume that the installed application is similarly faulty, even when this may not be true. Consequently, the team must do more than simply deploy; it must strive for a smooth deployment and prepare for the support and management of the product. This can include ensuring that training, infrastructure, and support are in place prior to deployment.

Team Model Role Clusters

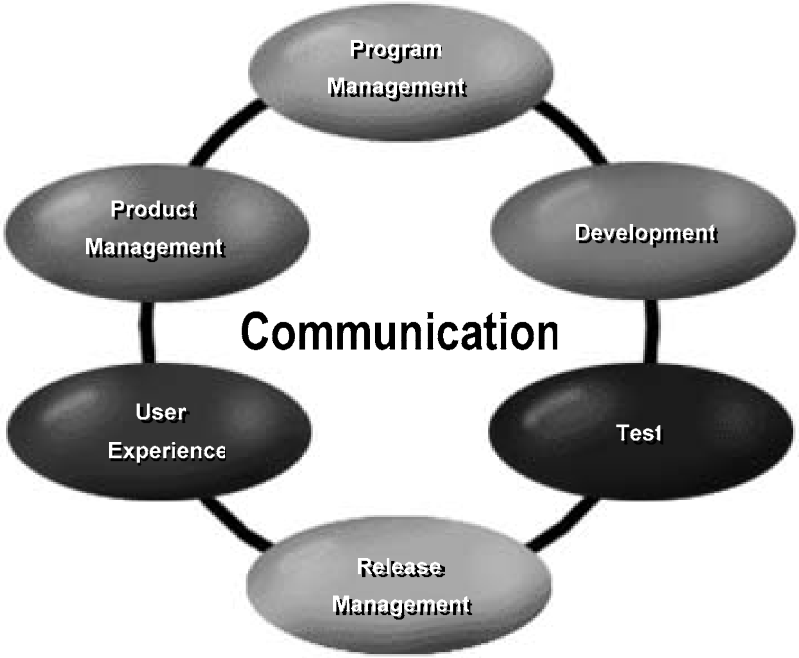

Figure 1 – Team Model Role Clusters

Product Management Role Cluster

The key goal of the product management role cluster is satisfied customers. Projects must meet the needs of customers in order to be successful. However, first the customer must be clearly identified and understood! In some cases the customer requesting a solution or set of features may be different from the sponsor who is paying or supporting effort. Thus there must be a clear distinction and requirements analysis for the success factors for both parties. Only then can the responsibilities of setting and meeting the expectations be assigned to the appropriate function areas. It is possible to meet budget and time goals but still be unsuccessful if customer and business needs have not been met.

The MSF team model separates functional areas for each role cluster in order to more narrowly define a set of responsibilities that when taken together often form a common skill set.

To achieve the goal of satisfied customers, the product management role cluster requires that several functional areas: product planning, business value, advocacy, and marketing.

Functional Areas

Marketing

- Drive marketing and public relations messages that have an impact on the target customer.
- Be highly differentiated so the solution stands out from the competition .
- Place the solution into distribution so that the target customer can easily acquire it.
- Provide support so that customers have a positive experience buying and using the solution.

Business Value

- Define and maintain the business justification for the project.
- Define and measure the business value realization and metrics.

Customer Advocate

- Drive a shared project and solution vision.
- Manage customer expectations and communications.

Product Planning

- Gather, analyze, and prioritize customer and business requirements.
- Perform market research, market demand, competitive intelligence/analysis.
- Determine business metrics and success criteria.
- Identify multi-version release plan.

Marketing

Marketing is the process or technique of promoting, selling, and distributing a product, solution, or service. There are several facets of marketing: Launch marketing, sustained marketing, and public relations. Over the course of a solution lifecycle, the focus of marketing will shift. Knowing the location of your solution within the lifecycle will be critical to executing the appropriate level of activities.

Business Value

Within the business value functional area, product management provides customers, Business Decision Makers (BDMs), with *as concise a predictive measure as they require* for the financial and operational return to the business from investment in an IT solution.

To be effective in providing a useful solution, product management must gain knowledge about customers business, success factors, and key performance measures. The process of capturing this knowledge can be defined as business value assessment or identifying critical success factors. Clearly, knowing what will make the customer successful helps in determining and proposing appropriate solutions. With increasing regularity, IT investments are coming under intense scrutiny and many IT-side contacts require financial review before signing off on projects. By performing objective cost-benefit analysis, the likelihood of satisfying the customer is increased. The calculation of financial results completes the development of a business case for IT investment.

Customer Advocate

This functional area contains responsibilities for high-level communications and management of customer expectations. High-level communications include public relations, briefings to senior management/customers, marketing to users, demonstrations, and product launches. Managing expectations is the key role of product management once the vision is set. It is considered to be a primary role because it can determine the difference between success and failure.

The importance of effectively managing expectations can be illustrated with an example involving the anticipated delivery of ten product features from a team to a customer by a certain date. If the team delivers only two features when the customer expects the delivery of all ten, the project will be deemed a failure both by the customer and by the team.

If, however, product management maintains constant two-way communication with the customer during the feature development and production period, changes can be made with regard to customer expectations that can ensure success. Product management can include the customer in the tradeoff decision-making process and inform them of changing risks and other challenges. Unlike the previous scenario, the customer can assess the situation and agree with the team that delivery of all ten features within the specified time frame is unrealistic and that delivery of only two is acceptable. In this scenario, the delivery of two features now matches the customer’s expectations and both parties will consider the project a success.

Product Planning

Product planning identifies the requirements and feature set(s) for multiple versions of a solution. A goal of product planning is to make it easy for a program manager or developer to understand a solution requirement in the least amount of time possible. This entails first, understanding the current requirements of a solution completely—what the needs of the business are, how customers will use it, what the support issues will be, and what alternatives are available. Second, the features that would add value to customers who use the solution are examined, such as the ability to enable entry into new business segments, integration with other systems, greater productivity, upgrading from other solutions, reducing support costs, and so on. Based on this knowledge, the product planner can recommend specific features that can be assigned to each solution release and prioritize the feature list.

At the core of product planning is research and analysis. Whether understanding the customer and business needs or understanding the competitive landscape, it comes down to appropriate attention to the research and analysis. This will prevent unnecessary features from being built into the solution.

Program Management Role Cluster

The focus of the program management role is to meet the goal of delivering the solution within project constraints. This can be viewed as ensuring that the project sponsor is satisfied with the outcome of the project. To meet this goal, program management owns and drives the schedule, the feature set, and the budget for the project. Program management ensures that the right solution is delivered at the right time and that the project sponsor's expectations are understood and managed throughout the project. Descriptions of selected functional areas are shown below.

Project Management

- Track and manage budget.
- Manage master project schedule.
- Drive risk management process.
- Facilitate communication and negotiation within the team.
- Track progress and managing project status reporting.
- Manage resource allocation

Solution Architecture

- Drive overall solution design.
- Manage the functional specification.
- Manage the solution scope and critical trade-off decisions.

Process Assurance

- Drive process quality assurance.
- Define and recommend improvements.

Administrative Services

- Implement the project management processes and support the team leads in using them.
- Provide a range of administrative services to support efficient team working.

Project Management

As the owner of the schedule, project management collects all team schedules, validates them, and integrates them into a master schedule that is tracked and reported to the team and the project sponsor.

As the owner of the budget, project management facilitates the creation of the project budget by gathering resource requirements from all of the roles on the team. Project management must understand and agree with all resource decisions (hardware, software, and people) and must track the actual expenditure against the plan. The team and the project sponsor receive status reports.

In addition, project management coordinates resources, facilitates team communication, and helps drive critical decisions.

For more information on the project management functional area and the MSF approach to the project management responsibilities, see: http://www.microsoft.com/msf

Solution Architecture

Solution architecture is the functional area of the program management role cluster responsible for the logical design of the solution and the functional specification. Solution architecture focuses on ensuring that a solution can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction.

Solution architect responsibilities include:

- Driving overall solution design.
- Managing functional specification.
- Managing the solution scope and critical trade-off decisions.

Owning the logical design, solution architecture provides the vital link between the business side of the solution (as represented by product management in the conceptual design) and the technology side of the solution (as represented by development in the physical design). Solution architecture acts as the custodian of the functional specification. It drives the team to achieve consensus about the content and design of the solution among the demands of their other roles, and justifies the agreed-on approach to the project stakeholders. It is also responsible for ensuring traceability of features back to requirements (and ultimately to the generation of business value), so that all features can be seen to support stated requirements and so that the team can assess the impact of any feature changes on the value of the solution.

Solution architecture activities include:

- Create the solution concept and align it with customer’s enterprise architecture; devise versioned release strategy; review plans for requirements capture.
- Capture requirements from architectural/standards groups and regarding interoperability; drive the logical design process; provide a traceability map tracing features back to requirements and benefits; create the functional specification; define interim releases.
- Manage changes to the functional specification; maintain traceability map; clarify the specification to other team roles and to external stakeholders; liaise with other project teams on interoperability issues.
- Participate in triage process; manage project stakeholders expectations regarding solution content.
- Provide updates to enterprise architecture team; update requirements for future versioned releases.

Solution architecture practitioners should be technically sound, with a broad base of knowledge and experience and the ability to relate the technical issues to the underlying needs of the business. While the solution architect may rely on the development team for expertise on the specific technologies being used in the solution, they must be able to grasp the implications of those technical details very rapidly and understand their inter-relationships and their impact on the environment into which the solution will be deployed. The solution architect must also be able to discuss those impacts with the customer’s architects so as to resolve rapidly any conflicts between the proposed solution and the enterprise architecture.

Process Assurance

The process assurance functional area of program management ensures that the project team adopts processes that focus on meeting the overall project quality goals, with an emphasis on eliminating sources of defect. Process assurance is responsible for two main areas:

- Defining the key project processes to be used by the team and providing advice and guidance to the team on their implementation
- Undertaking reviews to validate the relevance and effectiveness of the processes, recommending improvements, and monitoring compliance.

Process assurance focuses on the following activities:

- Define project protocols and processes in line with the project design.
- Provide advice and guidance on effective implementation of the project processes; validate compliance with processes; undertake milestone reviews; recommend process improvements.

Process assurance benefits from a degree of independence from the project team so that it can take an external perspective. For this reason, it is often managed from outside the project team, even if the project size does not make it a full-time role.

Administrative Services

This is the functional area of the program management role cluster that is responsible for implementation of the project management processes and for administrative support of the project team.

The project administration functional area ensures that the project team implements processes that meet the project design specification defined by project management. It is responsible for ensuring that the project team can operate effectively with the minimum of bureaucracy.

Project administration responsibilities include:

- Implementing the project management processes and supporting the team leads in using them. This includes consolidating team input to maintain the master plan and schedule and assisting leads in progress reporting.
- Providing a range of administrative services to support efficient team working, such as scheduling meetings, general procurement, and contract management.

Project administration focuses on:

- Supporting project initiation processes, such as efficient recruitment of team members from third parties; manage contractual arrangements; organize team facilities (space, telephones, security access, and so on).
- Establishing consistent planning framework; assist team leads in planning and scheduling; consolidate team input to create master plan and schedule; establish financial and progress reporting processes.
- Assisting team leads in progress reporting; create overall progress and financial reports.
- Ensuring closure of all administrative systems on project completion.
- Performing general administrative support activities such as scheduling meetings, implementing risk and issue management processes; maintaining the master risk list, action list, and so on; generating financial and progress reports; managing team location to enhance morale.

The project administration role requires a combination of strong administrative capability and attention to detail with sound experience in project planning and scheduling techniques, as well as a good understanding of the policies and guidelines operative in the supplier organization. On a larger project it provides an excellent opportunity to work alongside project direction and build the experience needed to direct future projects.

Development Role Cluster

The “build to specification” goal is the focus for the development role cluster during an MSF project. To succeed in meeting its quality goal, the role of development is to build a solution that meets the customer’s expectations and specifications as expressed in the functional specification. The development role cluster adheres to the solution architecture and designs that, together with the function specification, form the overall specifications of the solution.

In addition to being the solution builders, development serves the team as the technology consultant. As technology consultant, development provides input into design and technology selection decisions, as well as constructing functional prototypes to validate decision-making and mitigate development risks.

As builders, development provides low-level solution and feature design, estimates the effort required to deliver on that design, and then builds the solution. Development estimates its own effort and schedule because it works daily with all developmental contingency factors. MSF refers to this concept as bottom-up estimating, and it is a fundamental part of the MSF philosophy. Its goal is to achieve a higher quality of schedule and to increase accountability of those providing the estimates and of their work performance.

Technology Consulting Functional Area

- Serve the team as a technology consultant.
- Evaluate and validate technologies.
- Participate actively in the creation and review of the functional specification.
- Contribute to defining development standards for the organization.

Implementation architecture and design functional area

- Map the Enterprise Architecture (EA) to the solution’s implementation architecture by providing solution-specific detail for application, data, and technology views of the architecture.
- Own and implement the logical and physical designs of the solution.

Application Development Functional Area

- Code features to meet the design specifications.
- Conduct code reviews during development to share knowledge and experience.
- Carry out unit testing as defined in the test plan with the support of the test role.

Infrastructure Development Functional Area

- Develop features that meet the design specifications.
- Conduct code reviews during development to share knowledge and experience.
- Carry out unit testing as defined in the test plan with the support of the test role.
- Develop scripts for automated deployment.
- Develop deployment documentation.

Technology Consulting Functional Area

The technology consulting functional area serves as a technical resource throughout the project lifecycle. As a technology consultant, development must provide input into high-level designs, evaluate and validate technologies, and conduct research to mitigate development risks early in the development process.

During the envisioning phase, this functional area focuses on analyzing the requirements of the user/customer from an implementer's perspective. The functional area contributes to the definition of the vision/scope document by evaluating the technical implications of the project for implementation feasibility within the initial parameters of the project. It provides guidance on the pros and cons of possible implementation approaches and validates initial technology choices. In this process the functional area may conduct research, consult with counterparts in the organization or elsewhere, and hold discussions with technology providers. For additional validation, the functional area may develop a limited-functionality prototype to serve as a proof of concept. This is particularly relevant for projects that require the use of new technologies or in areas where the project team lacks experience.

Implementation Architecture and Design Functional Area

The implementation architecture and design functional area describes a set of responsibilities relating to the definition of an implementation architecture for the solution and the development of solution designs during an MSF project.

From a design standpoint, program management is responsible for the overall architecture of the solution and its positioning in the enterprise architecture. Development is responsible for mapping the enterprise architecture to the solution's implementation architecture by providing solution-specific detail for the application, data, and technology views of the solution.

MSF proposes a three-tiered design process: Conceptual design, logical design, and physical design. Program management and product management co-own conceptual design. Conceptual design includes user scenarios, high-level usability analysis, conceptual data modeling, and initial technology options. Development owns the logical and physical aspects of the solution design. Logical and physical designs require knowledge of relevant technology and the impact of technology choices on the design of a solution.

Application Development Functional Area

The application development functional area describes a set of responsibilities relating to the development of a software application during an MSF project. The development role's primary responsibility within this functional area is to build the features of the desired solution to specifications and designs, conduct unit testing, address quality issues identified in the testing process, and carry out the integration of solution components to produce the final deliverable.

The development role contributes to the definition of standards and adheres to these during solution development. Code reviews are conducted by development to assess the quality level of the application's features at the unit level. Reviews allow team members to share development knowledge and experience, supporting the MSF goal of "willingness to learn" for project teams. The development role is required to conduct and document results of satisfactory unit-level testing of the features implemented. The test role works actively with the development role to plan for and conduct the assessment of the quality of the solution feature independently and as part of the complete solution.

Infrastructure Development Functional Area

The infrastructure development functional area describes a set of responsibilities relating to the development of a systems and software infrastructure for a solution during an MSF project. The systems infrastructure includes the network infrastructure for a distributed computing environment, the client and server systems, and any supporting components. The software infrastructure includes the operating systems for clients and servers, as well as the software products that provide the required platform software services, for example, directory, messaging, database, enterprise application integration, systems management, network management, and so on.

During infrastructure development, the development role "develops" the infrastructure specified in the design. This includes configuring the foundation technology infrastructure for the solution, for example networking support, and the client and server systems as defined by the design. Aspects of the infrastructure can be influenced by the requirements of applications to be supported and vice versa. For example, a mission-critical high-performance solution may need to accommodate clustering and load-balancing of the back-end servers. Operating systems and platform products for the solution need to be appropriately "developed." The various software platform products must be installed, configured, and optimized to meet solution needs. After suitable testing and stabilizing, the infrastructure solution is deployed on a broad scale under the charge of the release management role, which has managed the acquisition of the solution's infrastructure requirements.

Test Role Cluster

The goal of the test role cluster is to approve for release only after all product quality issues are identified and addressed. All software is delivered with defects. A key goal is to ensure those defects are identified and addressed prior to releasing the product. Addressing can involve everything from fixing the defect in question to documenting work-around solutions. Delivering a known defect that has been addressed along with a work-around solution is preferable to delivering a product containing unidentified defects that may surprise the team and customer later.

To be successful, the test team role cluster must focus on certain key responsibilities. Those responsibilities are grouped within the three key functional areas.

Test Planning

- Develop testing approach and plan.
- Participate in setting the quality bar.
- Develop test specification.

Test Engineering

- Develop and maintains automated test cases, tools, and scripts.
- Conduct tests to accurately determine the status of product development.
- Manage the build process.

Test Reporting

- Provide the team with data related to product quality.
- Track all bugs and communicates issues to ensure their resolution before product release.

Test Planning

The test planning functional area is the part of the test role cluster that focuses on how the team will ensure that all product quality issues are identified and addressed.

The test role develops testing approaches and plans, and by doing so outlines the strategy the team will use to test the solution. These plans include the specific types of tests, specific areas to be tested, test success criteria, and information on the resources (both hardware and people) required to test.

An important part of the test planning functional area is participation in setting the quality bar by providing input to the project team on quality control measures and criteria for success of the solution.

The final activity within the test planning functional area is to develop the test specification. This is a detailed description of the tools and code necessary to meet the needs defined in the test plan.

Test Engineering

The test engineering functional area, as part of the test role cluster, focuses on carrying out the activities defined in test planning required to ensure that all product quality issues are identified and addressed. Among the responsibilities defined within this area are specific duties to develop and maintain test cases; development of tools, scripts, and documentation to perform testing functions; management of daily builds to ensure that test procedures can be performed and reported on a single frame of reference; and conducting tests to accurately determine the status of product development—running through the test cases, tools, and scripts to identify issues with the current build

Tracking and Reporting

The tracking and reporting functional area, as part of the test role cluster, focuses on articulating clearly to the project team what is currently wrong with the solution and what is currently right so that the status of development is accurately portrayed.

Issue tracking is performed to ensure that all identified issues have been resolved before product release. Document issue status, including assignment, priority, resolution, and work-arounds are completed on a frequent basis to provide the team with data related to current product quality status and detailed trend analysis.

User Experience Role Cluster

The goal of the user experience role cluster is enhanced user effectiveness. User experience is comprised of six functional areas: Accessibility, internationalization, technical communications, training, usability, and graphic design. The user experience role cluster has several responsibilities within each functional area that must be managed for the solution to be successful. Following is a listing of the functional areas and related responsibilities.

Accessibility

Drive accessibility concepts and requirements into design.

Internationalization

Improve the quality and usability of the solution in international markets.

Technical Communications

- Design and develops documentation for support systems (Helpdesk manuals, KB articles, and more).
- Document Help/assistance.

Training

Develops and executes learning strategy (build/buy/deliver).

Usability

- Gather, analyze, and prioritize user requirements.
- Provide feedback and input to solution design.
- Develop usage scenarios and use cases.
- Act as the user advocate to the project team.

Graphic Design

- Drives user interface design.

Accessibility

The accessibility functional area focuses on ensuring that solutions are accessible to those with disabilities by driving accessibility concepts and requirements into the design. Accessibility is important for many reasons. Primarily, accessibility is important because products and solutions need to be accessible and usable by all people regardless of their capabilities. A product or solution that does not account for accessibility will fall short of full adoption. Additionally, accessibility compliance will often be required to meet government regulations.

Accessibility concepts and requirements must be represented throughout the solution development cycle and should include:

- The incorporation of an accessibility section within each feature specification.
- Integrating accessibility information into the solution help section.
- Ensuring that accessibility documentation is complete.
- Ensuring that accessibility documentation is presented in accessible formats.

Internationalization

The responsibility within the internationalization functional area is to improve the quality and usability of the solution in international markets. The internationalization functional area is composed of both globalization and localization processes.

Globalization

Globalization is the process of defining and developing a solution that takes into account the need to localize the solution and its content without modification or unnecessary workarounds by the localizers. In other words, a released solution that is globalized properly is ready to localize with a minimum of difficulty.

Localization

Solution localization involves modifications to the solution’s user interface, Help files, printed and online documentation, marketing materials, and Web sites. Occasionally, these materials may require changes in graphical elements for a particular language version, or even content modifications.

Technical Communications

The technical communications functional area focuses on the development of solution document support systems.

A major responsibility of the technical communications functional area is the creation of tools components such as the Help tool. The Help tool empowers the user by providing answers to basic questions, keyword descriptions, error explanations, and frequently asked questions. Tools such as Help benefit both the user and the organization. Users benefit because they get responses to issues and questions in a timely and effective manner. The organization benefits by a reduction in support costs.

An additional responsibility of the technical communications functional area is designing and developing documentation for the solution. This may include the development of installation, upgrade, operations, and troubleshooting guides.

Training

The training functional area focuses on enhancing user performance by providing the skills knowledge needed to effectively use the solution. This skills knowledge transfer is achieved by implementing a learning strategy. The development of the learning strategy is the responsibility of the user experience team role cluster.

The development of the learning strategy may take place within the organization, or it may be outsourced to an organization that specializes in training and development. Regardless of who actually develops the learning strategy, the approach will most often consist of:

- Making an analysis of the user and the goals and objectives of the organization.
- Setting the desired skill level set.
- Developing and implementing a training plan.
- Upon implementation, measuring the training plan for effectiveness and modifying the training plan as appropriate, to assure success.

The learning strategy may comprise one or more of the following delivery mechanisms: Instructor-led training, technology delivered training, self study or the use of job aids. Many organizations choose a blended approach that adapts to the individuals own learning style.

Usability

The usability functional area focuses on ensuring that a solution can be used by specified users to achieve specified goals with high levels of effectiveness, efficiency, and satisfaction.

A major responsibility defined within the usability functional area is usability research, which includes gathering, analyzing, and prioritizing user requirements. By investing time to understand the user early on and throughout the solution development effort, the project will have a much higher likelihood of effectively meeting the needs of the users.

Another major responsibility as defined within the usability functional area is developing usage scenarios and use cases. The key idea here is to step back and look at how the entire solution will likely be used. This effort helps the development team understand how a user approaches the solution from a conceptual and literal standpoint and often will lead to design improvements resulting in increased efficiency.

The last major responsibility as defined within the usability functional area is providing feedback and input to the solution. By taking the time to provide user feedback to the developers throughout the development cycle, the solution will benefit by achieving a higher rate of user satisfaction.

Graphic Design

The graphic design functional area focuses on ensuring that graphical elements within the solution are designed appropriately. The major responsibility of the graphic design functional area is driving the design of the user interface. This involves designing the objects that the user is going to interact with (and the actions applied to those objects), as well as the major screens in the interface.

Release Management Role Cluster

The goal of the release management role cluster is smooth deployment and on-going operations. Release management is the role that directly involves operations on the MSF team. It includes the following functional areas of responsibility:

- Acts as primary advocate between project development and operations groups.
- Manages tool selection for release activities and drives optimizing automation.
- Sets operational criteria for release to production.
- Participates in design, focusing on manageability, supportability, and deployability.
- Drives training for operations.
- Drives and sets up support for pilot deployment(s).
- Plans and manages solution deployment into production.
- Ensures that stabilization measurements meet acceptance criteria.

Infrastructure

- Enterprise infrastructure planning.
- Coordinate physical environment use and planning across geographies (data centers, labs, field offices).
- Provide the team with policies and procedures for consistent infrastructure management and standards.
- Provide infrastructure services to the MSF team (building servers, standard images, installing software).
- Manage hardware/software procurement for the team.
- Build test and staging environments that accurately mirror production environments.

Support

- Provide primary liaison and customer service to the IT users.
- Support the business by managing the SLA with the customer and ensuring commitments are met.
- Provide incident and problem resolution; rapid response to user requests and logged incidents.
- Give feedback to development and design team.
- Develop failover and recovery procedures.

Operations

- Account and system setup controls; manage user accounts and permissions.
- Messaging, database, telecom operations; network operations.
- Systems administration, batch processing.
- Firewall management; security administration.
- Application services.
- Host integration services.
- Directory service operations.

Commercial Release Management

- Product registration codes; registration verification process.
- Licensing management.
- Packaging.
- Manage distribution channel.
- Print and electronic publication.

Infrastructure

The infrastructure functional area describes a set of responsibilities relating to the operations infrastructure that must be satisfied during an MSF project. It is part of the MSF release management role cluster. For projects using MOF, these correspond to the responsibilities of the MOF infrastructure role cluster.

Support

This functional area focuses on ensuring that the solution built and deployed is *supportable*. For projects using MOF, these correspond to the responsibilities of the MOF support role cluster.

Operations

This functional area describes a set of operations responsibilities that must be satisfied during an MSF project. This functional area focuses on ensuring that the solution built and deployed is *operable and compatible with other services in operation*. For projects using MOF, these correspond to the responsibilities of the MOF support role cluster.

Commercial Release Management

This functional area describes a set of responsibilities relating to releasing commercial software products. Commercial release management focuses on getting the product into the channel.

Scaling the Team Model

In his book *Rapid Development*, former Microsoft software developer Steve McConnell states:

"Large projects call for organizational practices that formalize and streamline communication. … All the ways to streamline communication rely on creating some kind of hierarchy, that is, creating small groups, which function as teams, and then appointing representatives from those groups to interact with each other and with management."

The MSF team model advocates breaking down large teams (those greater than ten people) into small, multidisciplinary feature teams. These small teams work in parallel, with frequent opportunities to synchronize their efforts.

In addition, function teams may be used where multiple resources are required to meet the needs of a particular role and are grouped accordingly within that role.

Feature Teams

Each role cluster in the team model comprises one or more resources organized in a hierarchical structure (although as flat as possible). For example, testers report to a test manager or lead.

Overlaid on this structure are feature teams. These are smaller sub-teams that organize one or more members from each role into a matrix organization. These teams are then assigned a particular feature set and are responsible for all aspects of it, including its design and schedule. For example, a feature team might be dedicated to the design and development of printing services.

Steve McConnell writes in *Rapid Development:*

> "Feature teams have the advantages of empowerment, accountability, and balance. The team can sensibly be empowered because it contains representatives … from each of the concerned parties. The team will consider all necessary viewpoints in its decisions and thus there will hardly ever be a basis for overriding it decisions.
>
> "For the same reason, the team becomes accountable. They have access to all the people they need to make good decisions. If they don't make good decisions, they have no one to blame but themselves. The team is balanced. You wouldn't want development, marketing, or quality assurance alone to have ultimate say over a product's specification, but you can get balanced decisions from a group that includes representatives from each of those categories."

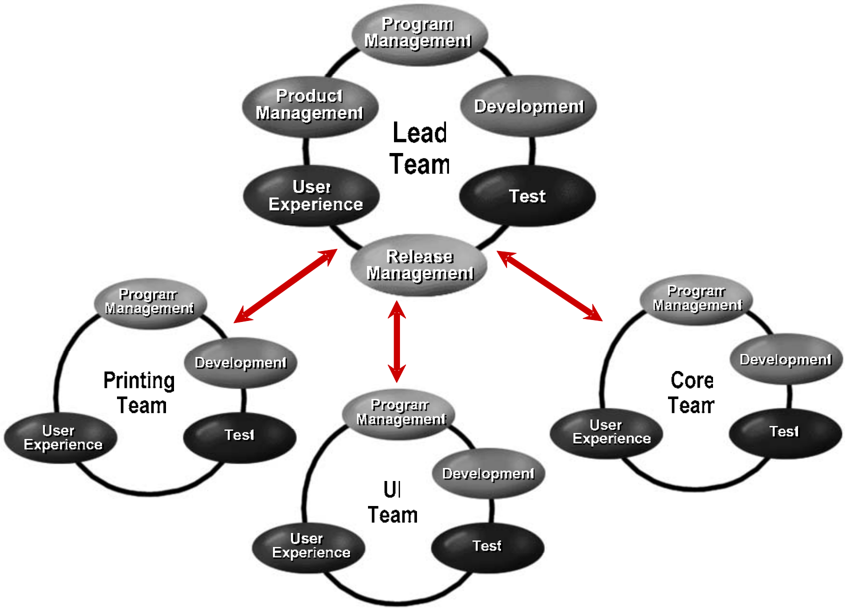

Figure 2 – Feature Teams

Note: This graphical example does not represent requirements for the organization of feature teams. For example, not all feature teams require the role of User Experience; the teams should be organized as required to meet the goal of their solution focus.

Function Teams

Function teams are teams that exist within a role. They are the result of a team or project being so large that it requires the people within a role to be grouped into teams based upon their functionality. For example, it is common at Microsoft for a product development team to have a product planner and a product marketer. Both jobs are an aspect of product management: One focuses on getting the features the customer really wants and the other focuses on communicating the benefits of the product to potential users.

This can also be true for development, where developers may be grouped by the service layer they work on: user, business, or data. It is also common for developers to be grouped on the basis of whether they are solution builders or component builders. Component builders are usually low-level C developers who create reusable components that can be leveraged by the enterprise. Solution builders build enterprise applications by “gluing” these components together. Solution builders usually work with higher level languages like Microsoft® Visual Basic®.

Often function teams include a hierarchical structure internal to that group. For example many program managers report up through lead program managers, with the leads reporting to a group program manager. A structure like this can also occur for the functional areas rather than at the role cluster level. The important thing to keep in mind is that the hierarchy does not hinder the team model at the project level. The goals of the roles remain the same as well at their overall accountability to the project team.

Sharing Roles

Even though the team model consists of six roles, a team doesn't need a minimum of six people. It also doesn't require one person per role. The key point is that six goals have to be represented on every team. Typically, having at least one person per role helps to ensure that someone looks after the interests of each role, but not all projects have the benefit of filling each role in that fashion. Often, team members must share roles.

On smaller teams, roles must be shared across team membership. Two principles guide role sharing. The first is that development team members never share a role. Developers are the project builders and they should not be distracted from their main task. To give additional roles to the development team only makes it more likely that schedules will slip due to these other responsibilities.

The second guiding principle is to try not to combine roles that have intrinsic conflicts of interest. For example, product management and program management have conflicting interests and should not be combined. Product management wants to satisfy the customer whereas program management wants to deliver on time and on budget. If these roles were combined and the customer were to request a change, the risk is that either the change will not get the consideration it deserves to maintain customer satisfaction or that it will be accepted without understanding the impact to the project. Having different team members represent these roles helps to ensure that each perspective receives equal weight. This is also true if trying to combine testing and development.

The following table illustrates risky (as indicated by "Not Recommended" or "Unlikely" symbols) and synergistic (as indicated by "Possible" symbols) combinations of roles, but as with any teaming exercise, successful role sharing comes down to the actual members themselves and what experience and skills they bring with them.

	Product Management	Program Management	Development	Test	User Experience	Release Management
Product Management		N	N	P	P	U
Program Management	N		N	U	U	P
Development	N	N		N	N	N
Test	P	U	N		P	P
User Experience	P	U	N	P		U
Release Management	U	P	N	P	U	

P = Possible U = Unlikely N = Not Recommended

Figure 3 - Combining Roles for Small Projects

The row column intersections with the N indicate that these roles are not recommended to be combined unless absolutely necessary because of conflicting interests and unless the associated risks can be addressed with associated risk mitigation and contingency plans. It is clear the goals of the roles have varying levels of conflict which both makes the team model dynamic and in turn increases the possibility of problems when trying to combine. That said, role combinations are not uncommon – and if the team chooses smart combinations and actively manages the associated risks, the problems that occur should be minimal.

Escalation and Accountability

The MSF Team Model Is NOT an Organization Chart

One question that often arises when applying the MSF team model is: “Who is in charge?” An organization chart describes who is in charge and who reports to whom. In contrast, the MSF team model describes key roles and responsibilities for a project team, but does not define the management structure of the team from a personnel administration perspective. In many cases, the project team includes members from several different organizations, some of whom may report administratively to a different manager.

There are certain situations that may arise, however, in which the team cannot come to consensus on an issue. After spending due diligence in trying to come to agreement, there are times that the role of program management must step up and take the primary lead in order to move the project forward. The primary goal of the program management role is delivery within project constraints, one which is time. Thus, to fulfill the goal of this role and of the team, there are times when the role of program management temporarily becomes a top down decision making authority in order to get the project back on track. It is during these instances that the leadership that has typically been shared throughout the roles understands the need for this shift and creates a stronger level of acceptance from the team and buy-in on the authoritative decision made for the purpose of reaching the project goals. As soon as the issue has been resolved and the team is able to get back into consensus there is an immediate shift back into the shared leadership responsibilities. The team of peers has proven to be flexible and adaptable enough to successfully handle these challenges yet remain a non-hierarchical approach to project teaming.

External Coordination—Who Is Accountable?

In order for a team to be successful, it must interact, communicate, and coordinate with other external groups. These range from customers and users to other development teams. In most cases, the customer requires explicit accountability for the solution to reside within one point of contact on the team. And although the team of peers requires a shared accountability within for the successful delivery of the solution, it is important to have a clear distinction of the accountability and reporting structure documented in the communications plan so that both the customer and the development team know how who on the team is responsible for facilitating this information.

The following diagram illustrates where responsibilities typically lie for coordination either with a business focus or a technology focus. Program management, product management, user experience, and release management are the primary facilitators. These roles are both internally and externally focused, whereas development and testing are internally focused and insulated from external communications.

This does not mean that developers and testers should be isolated from the outside world. Contact with the customer organization and with real users can be invaluable to build the customer-focused mindset that MSF teams look to achieve, especially in the earlier, formative stages of a project. Such communications should not, however, provide the formal communications since they would suffer badly as the development and testing teams focus on solution delivery during the latter stages of a project.

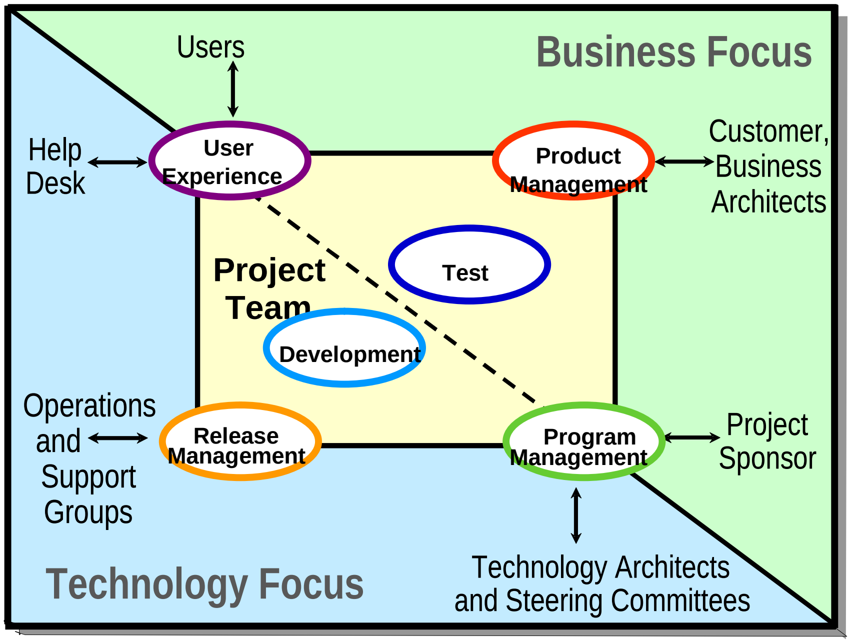

Figure 4 - Accountability

The diagram represents a high-level perspective. Typically, teams have to coordinate with many more external groups, such as quality assurance, finance, and legal. It is important that the interfaces with any external groups be explicit and understood and that development and testing continue to be insulated so that they can work effectively without unnecessary disruptions.

In addition it is important to emphasize that, while external coordination through the various roles can provide input and recommendations, neither individual members of the team or the team as a whole have the authority to change the priority or specifics of the project trade offs, such as features, schedule, and resources. Those changes are at the prerogative of the project customer or sponsor and implemented by the project team. This also provides an example of how a team of equal partners or peers defers to and aligns with organizational authorities, hierarchies, and structures.

Summary

The MSF team model is not a guarantee for project success. More factors than just team structure determine the success or failure of a project, but team structure is important.

In *Rapid Development*, Steve McConnell illustrates this point by saying:

"Even when you have skilled, motivated, hard-working people, the wrong team structure can undercut their efforts instead of catapulting them to success. A poor team structure can increase development time, reduce quality, damage morale, increase turnover, and ultimately lead to project cancellation."

The MSF team model is meant to address just that point. Proper team structure is fundamental to success, and implementing this model and using its underlying principles will help make teams more effective and therefore successful.

For more information, see:

Microsoft Solutions Framework: http://www.microsoft.com/msf

Microsoft Operations Framework: http://www.microsoft.com/mof

Endnotes

[1] See Jim McCarthy, *Dynamics of Software Development* (Redmond, WA: Microsoft Press, 1995) for more on optimizing team participation in design.