**Строгая методология разработки MSF (**Microsoft Solutions Framework)

Оглавление

[История создания 2](#_Toc128480882)

[Принципы, на которых базируется MSF 3](#_Toc128480883)

[Модели и дисциплины MSF 6](#_Toc128480884)

[Модель команды 6](#_Toc128480885)

[Команда проекта 7](#_Toc128480886)

[Роли и обязанности 7](#_Toc128480887)

[Модель процесса 10](#_Toc128480888)

[Управление проектом 12](#_Toc128480889)

[Управление рисками 13](#_Toc128480890)

[Управление готовностью 15](#_Toc128480891)

[Заключение 17](#_Toc128480892)

[Список литературы 18](#_Toc128480893)

# История создания

История создания MSF берет начало в 1993 году. Компания Microsoft уже была ведущей в IT-сфере и искала способы повысить эффективность и отдачу от своих проектов.

90-е годы стали временем расцвета новых подходов к разработке. Модель «Водопад», которую использовали больше двух десятилетий, уже в полной мере не отвечала требованиям в IT: была слишком жесткой и формализованной, медленно реагировала на новые потребности пользователей.

У Microsoft был обширный опыт в создании программных продуктов и продвижении масштабных IT-проектов: были уже выпущены Windows 3.11, Office 3.0 и многое другое. Компания суммировала накопленные знания и навыки, проанализировала опыт конкурентов и в 1993 году выпустила серию руководств, посвященных организации труда разработчиков — «белые книги» MSF.

Пять лет спустя, в 1998, была выпущена вторая ревизия MSF. В 2001 за ней последовала третья, а в 2005 вышла версия — MSF 4.0.

Отличительными чертами Microsoft Solutions Framework стали гибкость и масштабируемость. Эта методология подходит для работы в проектной группе или организации любого масштаба. MSF включает основополагающие принципы, модели и дисциплины управления персоналом, процессами и технологиями.

# 

# Принципы, на которых базируется MSF

Другими преимуществами методологии стали 8 основных принципов MSF:

* **Способствуйте открытому общению**

Модель MSF предусматривает свободный и открытый обмен информацией

между всеми членами команды и заинтересованными сторонами. Это

помогает исключить недопонимание между заказчиком и исполнителем и

снижает вероятность того, что работу придется переделывать. Все этапы

разработки обстоятельно описываются, обеспечивается доступность

документации для всех участников проекта — так налаживается

эффективное взаимодействие.

* **Работайте над общим видением**

Данный принцип Microsoft Solutions Framework подразумевает, что все

члены команды должны детально понимать цели и задачи, над которыми

работает коллектив. Общий взгляд на то, каким должен быть результат,

гарантирует, что усилия разработчиков будут согласованными.

Зачастую к успеху приводит именно умение правильно понять и

сформулировать достоинства выработанного решения. Кроме того,

эффективность труда в команде повышается, когда все сотрудники видят

картину проекта в целом. Это стимулирует их интересоваться даже теми

областями проекта, которые не относятся непосредственно к их задачам.

Благодаря этому сотрудники глубже вникают в принципы работы

программного продукта, обмениваются идеями и решениями, помогают

друг другу и нарабатывают новые компетенции и командный опыт.

Будьте готовы к переменам и оставайтесь гибкими

Любая разработка программного обеспечения требует времени. Это

означает, что на каком-то этапе работы требования заказчика могут

измениться — к этому стоит быть готовыми.

Все члены коллектива должны быть вовлечены в принятие решений об

изменениях в проекте. Благодаря этому новые задачи и цели будут

осмыслены и рассмотрены с учетом всех точек зрения и идей по

реализации.

* **Расширяйте полномочия членов команды**

Каждому члену команды должны быть предоставлены все полномочия,

необходимые для выполнения его обязанностей. Если его работа зависит

от коллег, он должен быть уверен, что с их стороны не будет задержек и

проволочек. Для дисциплины следует использовать графики, в которых

будут обозначены сроки для каждой задачи.

* **Разделяйте ответственность**

Все участники проекта принимают и осознают свою ответственность за

порученную задачи и собственные решения. В MSF проект делят на равноценные и уникальные сегменты. За успешную реализацию каждого в равной степени отвечают все специалисты, работающие над ним. Участник подотчетен своей рабочей группе, она —

всей организации, а та, в свою очередь, — заказчику. При этом

ответственность распределяется на каждом уровне равномерно между

всеми сотрудниками. Такой подход к разделению ответственности обусловлен тем фактом, что в общем решении зачастую сложно выделить вклад отдельного

специалиста. Согласно принципу, все успехи и неудачи проекта участники

делят поровну. Никто не может приписать себе единоличные заслуги или

стать козлом отпущения.

* **Сотрудничайте с клиентом и сосредоточьтесь на предоставлении бизнес-ценности**

Всегда нужно помнить о главном: программное решение должно

представлять ценность для бизнеса заказчика. MSF требует от команды

ориентироваться на клиента и вовлекать его в работу. В MSF — это значит понимать его цели и проблемы: зачастую заказчик совсем не разбирается в разработке программного обеспечения или даже в компьютерах, но это не мешает ему быть экспертом в своем бизнесе. Только он знает точно, каковы его потребности; какая функциональность

жизненно необходима, а какая избыточна; что имеет ценность для

бизнеса, а что — нет. Поэтому необходимо, чтобы клиент был вовлечен в

работу над проектом, и если он доволен результатами — все идет как

надо.

* **Будьте готовы к переменам и оставайтесь гибкими**

Любая разработка программного обеспечения требует времени. Это означает, что на каком-то этапе работы требования заказчика могут измениться — к этому стоит быть готовыми. Все члены коллектива должны быть вовлечены в принятие решений об

изменениях в проекте. Благодаря этому новые задачи и цели будут осмыслены и рассмотрены с учетом всех точек зрения и идей по реализации.

* **Инвестируйте в качество**

Успех команды зависит от того, насколько каждый специалист осознает

свою ответственность за продукт. Чтобы обеспечить высокое качество

решения, на протяжении всего проекта работает группа тестирования — в

ее задачи входит раннее выявление ошибок и недочетов. Обнаруженные

баги надо исправлять как можно скорее, чтобы они не повлияли на разработку. Microsoft Solution Framework требует, чтобы в планы и графики изначально вносилось время на устранение недостатков. Только в этом случае можно уложиться в срок.

* **Извлекайте хороший опыт**

Любой коллектив разработчиков, если только он не вчера появился,

обладает наработанным опытом. У каждого хорошего специалиста есть

свои испытанные временем приемы. Совокупные знания команды выходят

далеко за пределы компетенций отдельного сотрудника. Более того,

новый проект и каждая его итерация — это источник опыта. В коллективе важно создать и поддерживать среду, в которой каждый член команды чувствует себя востребованным, полезным и свободным. В психологически комфортных условиях возрастает не только эффективность сотрудника, но и его мотивация к самосовершенствованию и обмену знаниями с коллегами. В график работы необходимо закладывать время на обучение и общение — для этого в MSF существует дисциплина управления готовностью, о которой речь пойдет ниже. Периодически следует совместно анализировать выполненную работу. Необходимо поддерживать атмосферу доброжелательности и взаимопомощи. Открытость,

постоянный обмен информацией и опытом — залог успеха.

# Модели и дисциплины MSF

MSF разработана как комплекс отдельных компонентов — моделей и

дисциплин. Всего их пять, и они описаны в пяти «белых книгах» (white

papers) MSF.

Моделей используется две:

* модель команды;
* модель процесса.

А дисциплин в MSF три:

* управление проектами;
* управление рисками;
* управление готовностью.

## Модель команды

Команда разработчиков — главный стратегический ресурс компании,

определяющий успех проекта. В традиционной практике команды

организованы иерархически — от руководителя до работников низшего

звена, например:



*Пример традиционной иерархической организации труда на изображении выше.*

При такой организации работы вес мнения отдельного сотрудника

определяется не его компетенциями и знаниями, а положением в

иерархии. MSF предлагает более демократическую модель команды, и

поэтому не испытывает проблем классической.

### Команда проекта

Команда проекта в MSF — это коллектив равноправных сотрудников. Они

разделяют ответственность и свободно обмениваются опытом и

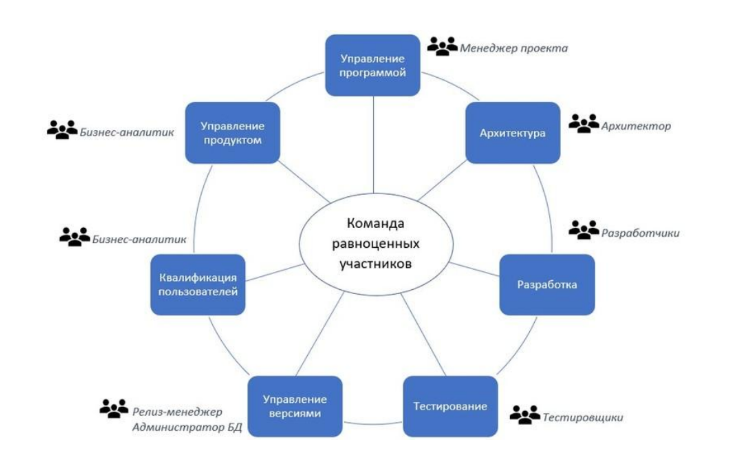
информацией. Внутри команды есть ролевые кластеры (роли),

отражающие функциональные обязанности конкретных специалистов. У

каждой роли — свои цели и задачи, и все они считаются равноценными и

одинаково важными. Роли дополняют друг друга и вместе служат единой

цели — созданию качественного продукта.



*Модель команды в MSF*

Эту модель можно масштабировать для работы как крупных компаний, так

и небольших коллективов.

### Роли и обязанности

* **Бизнес-аналитик**

Это главный посредник между командой разработчиков и клиентом. Он

должен детально разобраться в потребностях заказчика, определить

бизнес-ценность продукта и понять, какая именно функциональность

необходима в программе. Он трансформирует эту информацию в

конкретные определения и требования к качеству и доводит их до

разработчика. Можно считать, что бизнес-аналитик — представитель

клиента в коллективе. Он управляет продуктом, чтобы тот соответствовал

потребностям бизнеса и ожиданиям заказчика.

* **Менеджер проекта**

В модели MSF главная задача менеджера проекта — контролировать

график работ и бюджет. Он отвечает за то, чтобы все задачи выполнялись

своевременно и не выходили за пределы сметы. Занимается

планированием работы и составляет отчеты, оценивает риски и

вырабатывает меры по их снижению. Тесное сотрудничество с другими

ролями проекта позволяет ему быть в курсе событий и оперативно решать

административные проблемы.

* **Разработчик**

Кажется, что роль разработчика — самая незатейливая. Рабочая лошадка:

создает код и воплощает идеи клиента, трансформированные в

техническое задание усилиями бизнес-аналитика и архитектора. И

придерживается сроков, установленных проект-менеджером.

Но в MSF разработчик — полноправный участник всех обсуждений. Он

определяет, какое время потребуется ему для создания функциональных

блоков программы. Помогает архитектору выстраивать удачную структуру

проекта с точки зрения его реализации, основываясь на своем глубоком

знании языка программирования. Участвует в создании дизайна

приложения. В проекте считается экспертом по всем техническим

вопросам, и за ним зачастую остается последнее слово в совещаниях о

том, «как все это реализовать в рамках используемых технологий».

* **Тестировщик**

Задача тестировщика — выявить в продукте баги, проблемы и неудачные

решения, которые могут негативно сказаться на качестве и снизить

ценность приложения для клиента. Тестировщик обязан понимать и

учитывать контекст применения программного продукта: кто, как и для

чего будет его использовать на стороне заказчика. Если функция X

выполняется корректно и выдает правильный результат, но его получение

занимает в среднем час, тестировщик может признать ее работу

неудовлетворительной — зная, что клиенту требуется выполнять эту

функцию 20–30 раз в день.

Ошибки и отклонения от заданных параметров фиксируются и

документируются, после чего тестировщик передает их разработчику для

исправления. Тестировщик также участвует в выработке стандартов

качества и создании тестовых задач, нагрузочных тестов и подобного.

* **Релиз-менеджер**

В Microsoft Solutions Framework, что касается выпуска готового продукта

или любой его работоспособной версии, — на релиз-менеджере. Он

создает план выпуска версий, сертифицирует готовый продукт и следит за

тем, чтобы программа дошла до клиента. Кроме того, в его обязанности

входит информационное сопровождение версий, например описание

новых функций, изменений и нововведений.

При этом можно совмещать некоторые деятельности, ниже изображена таблица, которая позволяет определить, какие роли можно объединить.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Бизнес-аналитик | Менеджер программы | Разработчик | Тестировщик | Релиз менеджер |
| Бизнес-аналитик |  | - | - | + | -+ |
| Менеджер программы | - |  | - | -+ | + |
| Разработчик | - | - |  | - | - |
| Тестировщик | + | -+ | - |  | + |
| Релиз менеджер | -+ | + | - | -+ |  |

*Матрица совместимости ролевых кластеров MSF*

Анализируя данную таблицу, можно сделать следующие выводы:

* Недопустимо совмещать ролевые кластеры управления продуктом и управления программой.
* Ролевой кластер "Разработчик" нельзя совмещать ни с одним другим ролевым кластером.
* Совмещения других кластеров допустимы, но части из них - нежелательны.

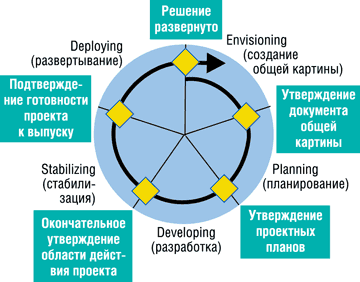
## Модель процесса

Водопадная модель предусматривает четкий переход от этапа к этапу: работы следующего этапа начинаются только после выполнения всех задач предыдущего. Такой стиль подходит для проектов, в которых проектные требования четко определяются заранее и с большой вероятностью не будут корректироваться потом. Данная схема организации разработки очень удобна с точки зрения управления проектом, так как позволяет четко сформулировать состав и обязанности его участников и контролировать графики выполнения проекта.

Спиральная модель обычно ориентируется на крайний случай, когда требования и параметры проекта непрерывно корректируются, а новые требования формулируются лишь по мере необходимости выполнения конкретных работ. Такая схема часто ассоциируется с понятием "экстремальной разработки"; при этом исполнитель и заказчик работают в постоянном тесном сотрудничестве, клиент привлекается на каждом этапе, формулируя свои соображения по поводу созданных компонентов. Однако при такой организации очень велик риск, что процесс разработки выйдет из-под контроля, поэтому реально данная модель используется лишь в относительно небольших проектах.

Однако проблема заключается в том, что чаще всего все требования на задание действительно практически невозможно определить заранее, к тому же даже сформулированные требования подвергаются коррекции. Но тогда требуется повысить уровень управляемости проектом, без чего создание сложного ПО просто невозможно. Компромисс между этими противоречивыми требованиями и предоставляет модель процессов MSF, в которой сочетаются водопадная и спиральная модели разработки: проект реализуется поэтапно, с наличием соответствующих контрольных точек, а сама последовательность этапов может повторяться по спирали (рис. 2).

В этом случае планирование на основе промежуточных контрольных точек и предсказуемость водопадной модели дополняются наличием обратной связи с заказчиком и творческим подходом к решению задач, характерными для спиральной модели. Таким образом, модель процессов MSF позволяет создавать и развертывать решения уровня предприятия, обеспечивая прогнозируемость хода проектов и учитывая реальные условия их выполнения.



*Этапы и контрольные точки модели MSF*

## Управление проектом

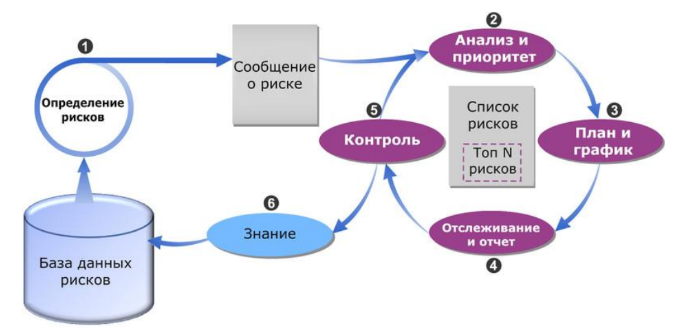
В MSF это целый набор навыков и компетенций, в том числе:

* комплексное планирование всех этапов и аспектов проекта
* управление бюджетом, расходами и ресурсами
* подготовка графиков и контроль за их соблюдением
* ведение административной документации.

Роль, ответственная за выполнение этого сегмента работы, — менеджер проекта (программы). По мере того, как масштаб проекта растет, управление проектом может разделиться на две специализированные ветви: одна будет связана с архитектурой программного решения и спецификациями, а другая — собственно с управлением проектом. Когда большие коллективы разрабатывают крупные проекты, управление может выполняться на нескольких уровнях: задачи распределяются между руководителями рабочих групп (команд, каждая из которых отвечает за ту или иную роль в проекте), а роль управления программой отвечает за координирование руководителей и в целом курирует проект. MSF стремится избавиться от иерархической структуры. Поэтому и при управлении проектом нет диктатуры. Демократичные обсуждения, при которых рассматриваются все точки зрения и достигается консенсус, способствуют выработке наиболее удачных решений. Когда члены команды не могут прийти к соглашению, менеджер проекта выступает арбитром: он обязан принять решение, максимально удовлетворяющее клиента и ориентированное на его бизнес-ценности.

## Управление рисками

Во избежание провала проекта данная тема была выделена в отдельный пункт, чтобы помочь выявлять, отслеживать и минимизировать риски.



*Модель управления рисками*

Она состоит из шести шагов:

1. **Определение рисков**

Любой член команды в любое время может (и должен) сообщать о рисках,

которые он выявил. Например, если обнаружил ошибку в сторонней

библиотеке кода, которая на данный момент не беспокоит, но грозит

привести к проблемам, когда будет использоваться соответствующая

функциональность библиотеки.

2. **Анализ и расстановка приоритетов**

Насколько серьезную угрозу представляет выявленный риск для проекта?

Возможно ли избежать проблем? Требуются немедленные действия или

время терпит? Необходимы ли дополнительные ресурсы, например время

на поиск решения?

Все эти вопросы обсуждаются коллегиально.

3. **План и график**

Информация, собранная при анализе рисков, должна быть преобразована

в конкретные планы, стратегии и действия. Следует сформулировать

четкие руководства, что необходимо делать и что запрещено, чтобы

снизить или исключить риски.

4. **Отслеживание и отчет**

Придерживаться принятого плана неукоснительно — половина дела в MSF.

Любое изменение в проекте может повлечь новые риски или рецидив

старых. Отслеживание рисков помогает держать их под контролем и

своевременно пересматривать тактику борьбы с ними.

5. **Контроль**

Контроль — это исполнение планов в отношении рисков и связанных с

ними отчетов.

6. **Знание**

Изучая риски, команда получает новую информацию о них и о способах преодоления сопутствующих сложностей. Эти знания необходимо фиксировать и помещать в базу данных, чтобы иметь к ним доступ в будущем.

## Управление готовностью

Эта дисциплина занимается вопросами профессионального роста и

подготовки специалистов.

С точки зрения организации, знания и навыки сотрудника — это ценный

ресурс, так что обучение и повышение квалификации можно

рассматривать как улучшение качества ресурса. В MSF под готовностью

понимается отношение текущего объема знаний и навыков к тому уровню,

который желателен или необходим для конкретного специалиста. От

готовности зависит, например, способность сотрудника выполнять ту или

иную роль в команде.

Для небольших или краткосрочных проектов подход к управлению

готовностью может быть простым — просто оценить знания сотрудников и

затем распределить роли в команде. Но организации, занимающиеся

долгосрочными проектами, могут извлечь из этой дисциплины

наибольшую выгоду, поскольку она предлагает комплексную программу

подготовки и повышения квалификации.

Процесс управления готовностью включает четыре этапа:

1. **Определение**

На этом этапе выстраивается структура команды. Для каждой роли

определяются уровни квалификации и компетенции, необходимые для

успешной работы специалистов. Кроме того, вырабатываются сценарии —

типичные виды деятельности, которые потребуются для разработки

проекта.

2. **Оценка**

Здесь внимание сосредоточено на каждом члене команды. Проводится

анализ компетенций и навыков, связанных с должностными

обязанностями, и определяется, насколько они соответствуют желаемым

показателям для каждой конкретной роли. Это позволяет выявить разницу

между текущим уровнем знаний и требуемым. В результате можно

разработать планы индивидуального обучения для каждого сотрудника,

которые позволят ему приобрести нужный уровень компетенций.

3. **Изменение**

В процессе обучения специалисты совершенствуют знания, чтобы

преодолеть разрыв между нынешним и желаемым уровнем квалификации.

При этом используются учебные планы со списками ресурсов и учебных

материалов — учебников, технических документов. Учебный план может

предусматривать и самостоятельное изучение, и под руководством

наставника.

4. **Подведение итогов**

На этом этапе проводится повторная оценка знаний и компетенций, чтобы

определить, были ли планы обучения эффективными и не требуются ли

дополнительные занятия. Рассматриваются не только теоретические

знания, но и способность сотрудника использовать их на практике.

Управление готовностью — процесс, в идеале не завершающийся на

протяжении всего проекта. Непрерывное совершенствование знаний и

умений каждого члена команды — путь к повышению качества и

успешности проекта в целом.

# Заключение

Методология MSF основана на гибкой процессной модели и включает в себя командную разработку. Масштабирование, команды команд – это достаточно важные составляющие MSF. Нужно сказать, что MSF поддерживает полный жизненный цикл разработки, т.е. он включает в себя как MSF (Microsoft Solutions Framework), так и MOF (Microsoft Operations Framework), которые объединяют процессы концептуализации, создания, внедрения, сопровождения, расширения, развития программных продуктов.

Данная методология хороша тем, что довольно демократичная по своей сути, а также подходит для больших и очень больших проектов, она гибкая и легко масштабируемая, четко определяются результаты по каждой контрольной точке. Не зря Microsoft уже почти четверть века является одним из мировых лидеров в сфере разработки программного обеспечения, что позволяет ей интегрировать новые продукты для массового пользования.

# Список литературы

1. Введение в методологию MSF – Андрей Колесов: URL: <https://bytemag.ru/vvedeniie-v-metodologiiu-microsoft-solutions-framework-1014/>
2. Алексей Березин – методологии разработки ПО: URL: <https://gb.ru/posts/development_methodology_msf>
3. Обзор методологий разработки корпоративных информационных систем: URL:

<https://science-engineering.ru/ru/article/view?id=1206>

1. Microsoft Solution Framework Team Model: URL: <https://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:Microsoft_Solution_Framework_Team_Model>