MSF (Microsoft Solutions Framework)

- *Каркас решений Microsoft* или *Фреймворк для создания решений от Microsoft* (МСФ, MSF).
- МСФ представляет собой каркас процессов, основанных на принципах и использующих опробованные практики.

MSF (Microsoft Solutions Framework)

- Microsoft Solutions Framework является также продуктом, предоставляемым Microsoft. В этом качестве он представляет собой базу знаний в виде пакета руководств, разделённого на несколько *белых книг* – документов, каждый из которых охватывает определённую модель или дисциплину. Он входит в набор инструментальных средств Microsoft Visual Studio Team System для поддержки МСФ.

MSF (Microsoft Solutions Framework)

МСФ основан на наборе
из 9 основополагающих принципов,
формирующих общую суть моделей
и дисциплин:

1. Работа в рамках единого видения;
2. Проявление живости, ожидание изменений;
3. Сотрудничество с заказчиками;
4. Поощрение свободного общения;
5. Обучение на любом опыте;
6. Вкладывание [денег] в качество;
7. Поставка инкрементного результата;
8. Установление ясной подотчётности;
9. Наделение полномочиями членов команды.

MSF (Microsoft Solutions Framework)

МСФ предлагает набор из ключевых концепций:

1. Фокусировка на конечном результате;
2. Поддержка своей клиентуры;
3. Чувство гордости за мастерство;
4. Просмотр всей картины;
5. Непрерывное обучение;
6. Усвоение качества обслуживания.

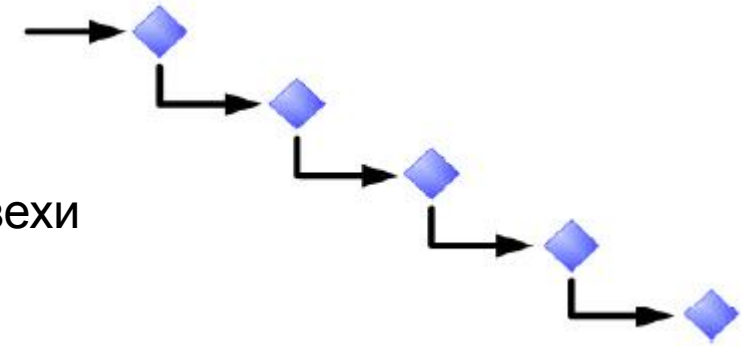
MSF (Microsoft Solutions Framework)

Модель руководства МСФ обладает следующими тремя особенностями:

1. Итеративный подход;
2. Подход, основанный на фазах и вехах;
3. Целостный подход к созданию и внедрению решений.

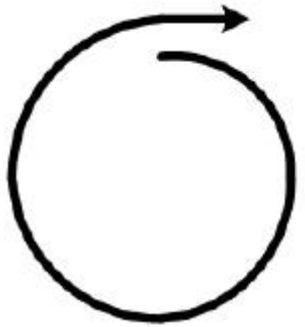
MSF (Microsoft Solutions Framework)

Модель процесса MSF

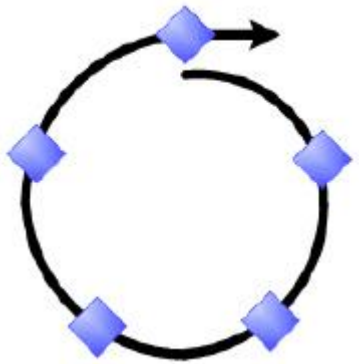


Водопадная модель – фазы *работ* и вехи

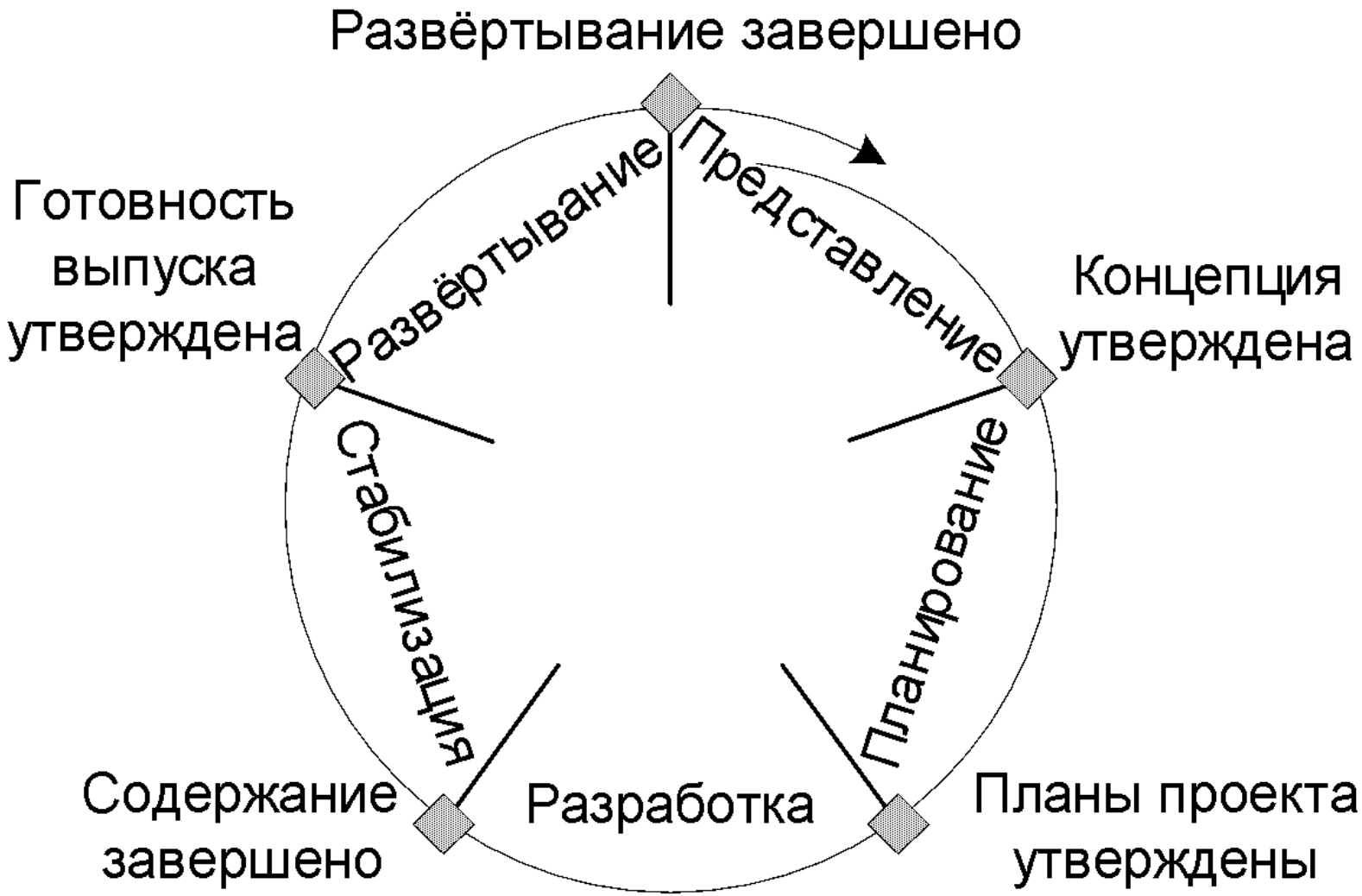
Спиральная модель – постоянное уточнение требований, активное взаимодействие с заказчиком.



В MSF объединяются водопадная и спиральная модели: сохраняются преимущества упорядоченности водопадной модели, не теряя при этом гибкости и творческой ориентации спиральной модели.



Модель ЖЦ для подхода MSF



Модель ЖЦ для подхода MSF

Модель ЖЦ для МСФ отражает один цикл разработки

- В МСФ выделено всего 5 фаз:
  1. Представление;
- 2. Планирование;
- 3. Разработка;
- 4. Стабилизация;
- 5. Развёртывание.

Все фазы разграничены главными вехами. Для повышенного управления проектом внутри фаз выделяют ряд промежуточных вех, показывающих достижение результата в некоторой деятельности.

- На фазе 1 выполняется создание и сплочение команды на основе выработки единого видения. Основными задачами являются создание ядра команды и подготовка документа с описанием концепции проекта, включающего видение и содержание проекта. Главная веха 1 считается достигнутой, если команда и заказчик пришли к соглашению об общих задачах и сроках проекта, включаемой и не включаемой в решение функциональности. Результатами являются: Описание видения и содержания, Документ оценки рисков, Описание структуры проекта.

- На фазе 2 производится основная работа по составлению планов проекта. Она включает в себя подготовку командой функциональной спецификации, разработку дизайнов, подготовку рабочих планов, оценку проектных затрат и сроков разработки различных составляющих проекта. Главная веха 2 считается достигнутой, если заказчик и команда пришли к соглашению о составе решения и сроках поставок. Утверждённые спецификации, планы и календарные графики образуют *базовый план проекта*. Результатами являются: Функциональная спецификация, План управления рисками, Сводный план и сводный календарный график проекта.

- На фазе 3 команда фокусируется на создании компонентов решения. Некоторая часть этой работы может продолжаться на следующей фазе, если такая необходимость выявлена при тестировании. Эта фаза также включает в себя разработку инфраструктуры. Главная веха 3 считается достигнутой, если создание всех компонентов решения завершено и решение готово к тестированию и стабилизации. Результатами являются: Исходный и исполнимый код приложений, Скрипты установки и конфигурирования, Окончательная функциональная спецификация, Материалы поддержки решения, Спецификации и сценарии тестов.

- На фазе 4 производится тестирование разработанного решения. При этом внимание фокусируется на его эксплуатации в реалистичной модели производственной среды. Главная веха 4 считается достигнутой, если команда завершила разрешение всех существенных проблем и производится выпуск или внедрение решения. Результатами являются: «Золотой» выпуск, Документация выпуска, Материалы поддержки решения, Результаты и инструментарий тестирования, Исходный и исполнимый код приложений, Проектная документация, Обзор вехи.

- На фазе 5 команда внедряет технологии и компоненты решения, стабилизирует внедрённое решение, передаёт работу персоналу поддержки и сопровождения и получает со стороны заказчика окончательное одобрение результатов проекта. По завершении внедрения команда производит анализ работы и удовлетворённости заказчика. Главная веха 5 считается достигнутой, если решение начало давать заказчику результат, а команда может свернуть свою деятельность. Результатами являются: Информационные системы эксплуатации и поддержки, Процедуры и процессы, Базы знаний, отчёты, журналы протоколов, Версии проектных документов, массивы нагрузки и код, разработанные во время проекта, Отчёт о завершении проекта, Окончательные версии всех проектных документов, Показатели удовлетворённости заказчика и пользователей, Описание последующих шагов.

- Следует сделать следующие замечания по этой модели ЖЦ. Длительность фаз не одинакова. Деятельность может выходить за рамки одной фазы. Наличие / отсутствие некоторых фаз определяется выполняемым проектом.
- Таким образом, МСФ предлагает модель ЖЦ, основанную на распределении работ в команде проекта по фазам, а не на выделении процессов.

Agile
(гибкая методология разработки ПО)

В отрасли разработки программного обеспечения термин Agile Development Method появился в начале 2000-х годов, когда в штате Юта был издан «Манифест гибкой разработки ПО».

С тех пор под «Agile» понимают набор подходов по “гибкой” разработке программного обеспечения.

15

Agile
Манифест «живой» разработки

Основные принципы «живой» разработки ПО зафиксированы в манифесте «живой» разработки:

- Люди их общение более важны, чем процессы и инструменты
- Работающая программа более важна, чем исчерпывающая документация
- Сотрудничество с заказчиком более важно, чем обсуждение деталей контракта
- Отработка изменений более важна, чем следование планам

16

Agile
Манифест «живой» разработки

Для заказчика манифест можно коротко сформулировать так:

- разработка ведется короткими циклами (итерациями), продолжительностью 1-4 недели;
- в конце каждой итерации заказчик получает ценное для него приложение (или его часть), которое можно использовать в бизнесе;
- команда разработки сотрудничает с Заказчиком в ходе всего проекта;
- изменения в проекте приветствуются и быстро включаются в работу.

17

Agile
методологии гибкой разработки ПО

Наиболее известные методологии гибкой разработки ПО:

- Scrum;
- Lean;
- Feature Driving Development;
- Extreme Programming (XP).

18