**Azure DevOps Server** (ранее **Team Foundation Server**, сокр. TFS) — продукт корпорации Microsoft, представляющий собой комплексное решение, объединяющее в себе систему управления версиями, сбор данных, построение отчётов, отслеживание статусов и изменений по проекту и предназначенное для совместной работы над проектами по разработке программного обеспечения. Продукт доступен в виде отдельного приложения, сходного по функциям с облачным сервисом Azure DevOps Services (до 2019 года называвшимся Visual Studio Team Services, VSTS).

1. **Архитектура.**
2. Team Foundation Server работает по трёхуровневой архитектуре: *клиентский уровень*, *прикладной уровень* и *уровень данных*. *Клиентский уровень* используется для создания и управления проектами, а также для доступа к хранимым и управляемым элементам проекта. На этом уровне TFS не содержит никаких пользовательских интерфейсов, но предоставляет веб-сервисы, которые могут быть использованы клиентскими приложениями для самостоятельной интеграции в функциональность TFS. Эти веб-сервисы используются такими приложениями, как Visual Studio Team System для применения TFS в качестве серверной инфраструктуры хранилища информации или выделенного TFS управления приложениями, наподобие включенного приложения *Team Foundation Client*. Сами веб-сервисы находятся на *прикладном уровне*. *Прикладной уровень* также включает в себя веб-портал и репозиторий (хранилище) документации, поддерживаемые Windows SharePoint Services. Веб-портал, называемый *Team Project Portal* (портал командного проекта), выступает в роли центра взаимодействия для проектов, управляемых TFS. Репозиторий документов используется как для элементов проекта, так и для отслеживания ревизий (документирование изменений), а также для накопления и обработки данных и генерации отчётов. *Уровень данных*, основывающийся в первую очередь на установленном SQL Server 2005 Standard Edition, обеспечивает сервисы постоянного хранения данных для репозитория документов. *Уровень данных* и *уровень приложений* могут существовать на различных физических или виртуальных серверах при использовании Windows Server 2003 или более специализированных версий. *Уровень данных* не взаимодействует с *клиентским уровнем* напрямую, только через *прикладной уровень*.
3. Большая часть действий в Team Foundation Server происходит с «рабочими элементами». Рабочими элементами называются отдельные единицы (шаги) работы, выполняемые поочерёдно. Во многих источниках они отождествляются с элементами типа «ошибка» (bug) в системах отслеживания ошибок наподобие Bugzilla, то есть в этом случае рабочий элемент имеет поля Area (связанная область), Iteration (состояние), Assignee (связан с), Reported By (кем создан) для указания соответствующей информации, историю, приложенные файлы, а также множество прочих атрибутов. Рабочие элементы сами по себе могут быть нескольких типов, например *Ошибка*, *Задача*, *Требование качества*, *Сценарий* и т. д. Выбранный фреймворк для любого взятого проекта в Team Foundation Server определяет какие типы рабочих элементов будут доступны и какими атрибутами каждый из типов будет обладать. Эти элементы внутри приложения хранятся в XML-формате, а их схема может быть легко модифицирована для добавления новых атрибутов к различным элементам или создания новых элементов на проектной основе. Каждый рабочий элемент имеет соответствующие методики контроля, которые определяют, кому какие именно элементы доступны и какие действия он с ними может производить (просмотр, редактирование, создание, удаление и др.). Также подразумевается использование уведомлений и возможности ведения логов, для отслеживания истории всего процесса создания, обращения к элементу или его изменения (определяется правами), а также предусматривается дополнительное уведомление определённых пользователей при наступлении определённых событий.
4. Любой Team Foundation Server содержит один или более *Совместный проект* (Team Projects), состоящий из решений на базе Visual Studio, конфигурационных файлов для Team Build и Team Load Test Agents, и единый репозиторий на базе SharePoint, содержащий связанную с проектом документацию. Совместный проект включает в себя пользовательские рабочие элементы, версии (ветки) исходного кода, отчёты, управляемые TFS. TFS обеспечивает возможности для управления этими проектами. При создании нового проекта нужно выбирать фреймворк программной разработки, который впоследствии сменить нельзя. TFS включает в себя несколько наиболее распространённых шаблонов, среди которых присутствуют как методики гибкой разработки, так и формальные. Выбор фреймворка наполняет проект заранее предопределенными элементами, как например, роли и полномочия, а также прочую документацию, например, стратегию развития проекта (project roadmap), шаблоны документов, заготовки отчётов. Эти элементы могут быть связаны с рабочими элементами. Статус определённых элементов проекта может автоматически обновляться при изменении рабочих элементов. TFS можно интегрировать с Microsoft Excel для создания и отслеживания элементов проекта. Статус элементов в этом случае можно указывать и редактировать непосредственно в Excel, а итоговые таблицы могут обрабатываться TFS, который будет импортировать данные с учётом особенностей управления данным проектом. Кроме того, его также можно интегрировать с Microsoft Project (например, Microsoft Project 2003, но не Project Server!) в качестве клиентской части управления проектом. Элементы проекта можно экспортировать как документы Excel для дальнейшего анализа данных.
5. TFS сам по себе не содержит пользовательского интерфейса для выполнения подобных заданий. Подобные возможности предоставляются за счёт веб-сервисов, которые используются клиентскими приложениями наподобие среды разработки Visual Studio Team System (VSTS). Тем не менее, TFS имеет в своём составе приложение *Team Foundation Client* (TFC), которое может использоваться для выполнения этих заданий без VSTS. TFC также оперирует вызовами соответствующих веб-сервисов. TFS предоставляет клиентский API, который может быть использован клиентским приложением для доступа к функционалу; сам API управляет промежуточными связями для установки взаимодействия с веб-сервисами, как, например, кэширование со стороны клиента для уменьшения задержек и накладных расходов. Также поддерживается язык описаний веб-сервисов WSDL на случай если приложению требуется непосредственный вызов веб-сервисов. В качестве дополнения доступно Visual Studio Team System Web Access, предназначенное для схожих целей.
6. **Контроль исходного кода (Source control).**

Team Foundation Server реализует репозиторий управления исходным кодом, называемый *Team Foundation Version Control* (TFVC). В отличие от предыдущего решения Microsoft по контролю кода, Visual SourceSafe (VSS), которое основывалось на механизме файлового хранения, Team Foundation хранит весь код, равно как и запись обо всех изменениях кода в базе данных под управлением SQL Server. Поддерживаются такие особенности, как например, одновременная *множественная блокировка кода для изменения* (multiple simultaneous check-outs) (то есть один и тот же файл одновременно могут редактировать несколько человек), разрешение конфликтов, откладывание внесения изменений (shelving) (здесь имеется в виду сохранение набора запланированных изменений без их подтверждения в системе контроля версий, причём другие пользователи могут видеть эти наборы, но доступа к ним без явно предоставленного разрешения не получат), ветвление и слияние, а также возможность устанавливать уровни доступа (безопасности) на любом уровне дерева исходного кода, наряду с наиболее очевидными возможностями отслеживания версий документации, блокировок, откатов и операций подтверждения микроизменений (atomic commits). Механизм контроля исходного кода напрямую связан с рабочими элементами Team System; При подтверждении изменений (check-in) (иначе называемом ещё «набором изменений» (changeset)) разработчик может определять взаимосвязь его кода с одним или более определёнными рабочими элементами для указания какие именно проблемы решает данное подтверждение (check-in). Администраторы TFS могут принудительно указывать политики подтверждений (check-in policies), которые предусматривают удовлетворение требованиям Code Analysis, а кроме того можно поставить обязательное указывание рабочих элементов, связанных с данным подтверждением, или обновление статуса связанных рабочих элементов (как например, указание ошибки как «исправленной» при внесении изменений в код, исправляющих данную ошибку). Отдельные версии файлов могут отмечать специальными метками, а все файлы с одинаковыми метками образуют релиз-группу. В отличие от VSS, репозиторий контроля кода TFS не обладает ни поддержкой привязки к элементу из нескольких мест структуры каталогов исходного кода, ни поддерживает «закрепления» (pinned) элемента (то есть поддержка различных ссылок на один и тот же файл из нескольких каталогов для различных версий чтобы избежать дальнейших правок этого файла).

TFVC поддерживает ветвление на всех уровнях исходного кода, а также для отдельных файлов и каталогов, причём каждая ветвь поддерживается отдельно. Несколько ветвей можно объединить в одну с указанием порядка (алгоритма) разрешения конфликтов при слиянии изменений двух ветвей одного файла, тогда программа сама автоматически согласует различия или отметит их для ручной проверки, если сама с ними не справится. Слияние может быть выполнено и на уровне набора изменений («changeset»), вместо уровня ветвления. Успешное слияние автоматически отмечается (check out) в репозитории контроля кода.

Возможности TFVC не ограничены лишь исходным кодом, с помощью встраивания инфраструктуры Windows SharePoint Services он обеспечивает поддержку библиотеки версий документов проекта, включая планы проекта, требования, анализ специфики проекта и другие. Все документы в репозитории контроля кода могут связываться с любым рабочим элементом, а доступ к ним может контролироваться за счёт введения политики доступа (ограничения прав).

1. **Отчетность(Reporting).**

Reporting — ещё один основной компонент Team Foundation Server. При помощи него можно создавать множество отчётов на основе объединения информации о рабочих элементах, наборах изменений, информации, поставляемой Team Build, и результатов тестирования от Test Agents. Например, уровень изменений кода за определённый временной промежуток, списки ошибок, не имеющих тестовых наборов, повторения ранее пройденных тестов и т. д. Отчёты, созданные при помощи SQL Server Reporting Services, можно экспортировать в нескольких различных форматах, включая Excel, XML, PDF и TIFF. Отчёты можно просматривать как при помощи Visual Studio, так и через веб-портал.

TFS использует свой фреймворк логирования для автоматизации сбора данных. Инфраструктура логирования отслеживает и записывает информацию, касающуюся доступа и использования рабочих элементов и исходного кода, которая затем может использоваться сервисами анализа для выявления направлений (трендов). TFS на уровне данных содержит адаптер накопления, кэширующий данные из нижележащих нормализованных баз данных в удобной форме для анализа — таблицы или размерные таблицы. Затем для анализа этих данных используется SQL Server Analysis Services, и создаются отчёты. Отчёты могут охватывать несколько рабочих элементов, включая основные направления ошибок, изменения кода, направления сборок и другие. Другие аналитические приложения также могут использовать данные напрямую предоставляемые веб-сервисами.

### Портал проекта (Project portal).

### Исходя из проектной основы, TFS также создаёт SharePoint-сайт для проекта, который может использоваться для отслеживания прогресса проекта, наблюдения за рабочими элементами и документами, представленными в библиотеке проекта. На сайте также можно просматривать созданные отчёты. TFS можно применять в качестве центра связи, то есть пользователи, связанные с определённым проектом, могут использовать сайт для общения или взаимодействия друг с другом. Комментарии могут связываться с различными элементами. Для каждого проекта, в зависимости от его свойств, TFS использует заранее предопределённые шаблоны, указываемые при создании сайта. Такие шаблоны могут настраивать (редактировать) администраторы TFS.

1. **Shared services.**

TFS обеспечивает поддержку множества сервисов, которые могут быть использованы для интеграции с посторонними приложениям, как например, IDE и системы управления проектами. *Сервис связывания* (linking service) позволял создавать слабосвязанные отношения между элементами, например элемент *ошибка* и версии исходного кода, связанные с ним. *Сервисы безопасности* (security services) позволяли создавать среди пользователей группы безопасности, которым устанавливались права доступа. *Сервис классификации* (classification service) предусматривал определение политик автоматической классификации элементов исходя из множества критериев, а *Сервис событий* (eventing service) позволял любому компоненту вызывать событие и уведомление, связанное с этим событием. Уведомление может происходить как при помощи подписки на поток определённой информации, так и при помощи электронной почты или путём вызова других веб-сервисов.

1. **Team Build**

*Team Build* — сервер сборки, входящий в состав Team Foundation Server, и который может быть установлен практически на любой машине, поддерживающей Visual Studio. Машины, сконфигурированные под Team Build, могут использоваться разработчиками для выполнения полной сборки большинства последних версий программного обеспечения, используемых в контроле кода. Записи каждой сборки сохраняются вне зависимости от её успешности или неуспешности, так что разработчики и администраторы сборок могут отслеживать прогресс проекта. Если сборка происходит последовательно, то анализируются изменения, сделанные в исходном коде после последней успешной сборки, а обновление рабочих элементов указывает на определённый прогресс. Например, если тестировщик заводит рабочий элемент, посвящённый конкретной ошибке в сборке #15, а разработчик вносит изменения чуть ранее, чем была создана сборка #18, то элемент «ошибка» обновится до статуса, указывающего, что ошибка исправлена. Тестировщик может как подтвердить, так и опровергнуть то, что ошибка была успешно исправлена.

На данный момент существуют две версии TeamBuild, причём каждая версия соответствует устанавливаемой версии TFS. Впрочем, они вполне легко настраиваются.

TFSBuild.proj — файл, управляющий TeamBuild. Язык Team Build схож с языком MSBuild