**Проблемы разработки ПО**

**Проблемы**

При разработке ПО часто срываются графики работ и наблюдается превышение установленного бюджета. Часто поставленный программный продукт не отвечает требованиям потребителя и его никогда не используют. Программные продукты часто просто плохо работают. Как видно из симптомов недомогания ПО, при его разработке возникает пять принципиальных вопросов:

Недостаток прозрачности;

Недостаток контроля;

Недостаток трассировки;

Недостаток мониторинга;

Неконтролируемые изменения.

|  |  |
| --- | --- |
| Вопрос | Ответ |
| Недостаток прозрачности | ПО по своей природе является концептуальным. В отличие от моста, здания или любого другого физического объекта, сложно посмотреть на программный продукт и оценить степень его завершенности. Без жесткого руководства проектом разработка ПО будет завершена на 90% при использовании 90% отведенного времени. Политика Управления Конфигурациями (УК)и Управления Изменениями (УИ) и определение модели менеджмента конфигурации ПО при разработке продукта, все элементы конфигурации, компоненты и подкомпоненты мгновенно становятся видимыми для версий, релизов и семейств продуктов. |
| Недостаток контроля | Поскольку программное обеспечение является нематериальным в физическом смысле, его более сложно контролировать. Без точной оценки процесса разработки срываются графики выполнения работ и превышаются установленные бюджеты. Очень сложно оценить объем выполненной и оставшейся работы. Процесс УК и УИ предоставляет механизм управления процессом через определение фактически затраченных и плановых ресурсов и оценивание будущих затрат, исходя из объема выполненной работы. |
| Недостаток трассировки | Отсутствие связи между отдельными событиями проекта приводит к его провалу. Главное преимущество УК и УИ состоит в том, что с его помощью обеспечивается трассировка среди версий, релизов и семейств продуктов. Ценность подобной трассировка огромна в ситуациях, когда в одном из выпусков или семействе продукта возникает проблема, которая оказывает влияние на другие клиентские релизы и продукты. Выполнение одного изменения и его распространение на всю базу ПО экономит много времени, средств и улучшает взаимоотношения с клиентами. Отсутствие связи между событиями проекта приводит к его провалу, когда решение одной проблемы увеличивают проблему в другой области или приводит к неудаче в попытке решить аналогичную проблему где то в другом месте. Сквозная трассировка выполняемых задач позволяет менеджменту в пределах аудиторской возможности УК и УИ проверить цепочку событий, из за которых возникли сложности в проекте как интегральном процессе. А отслеживание календарного графика выполнения работ позволяет, не затягивая проект, завершать разработку ПО в установленные сроки. |
| Недостаток мониторинга | Без трассировки и «прозрачности» сложно осуществить мониторинг программных проектов. Руководство не может принять компетентные решения, поэтому графики продолжают срываться, а затраты продолжают превышать установленный бюджет. Невозможно выполнить мониторинг проекта, если у менеджера проекта нет инструментальных средств, чтобы следить за фактической разработкой продукта в пределах проекта. В ходе осуществления процесса УК и УИ реализуется обеспечение инструментальными средствами, которые позволяют осуществить разносторонний мониторинг процесса. При наличии УК и УИ, трассировки и «прозрачности» мониторинг программных проектов становится простой частью общей задачи управления проектом. С помощью мониторинга, доступного благодаря инструментальным средствам УК и УИ и возможностям CCB\*, менеджеры проектов принимают взвешенные решения, не выбиваясь из графика работ и не превышая бюджет. |
| Неконтролируемые изменения | ПО является достаточно гибким, оно представляет результат работы большого коллектива, но у потребителей постоянно возникают новые идеи относительно данного программного продукта. Люди редко просят конструктора моста внести изменения в середине проекта, тогда как пользователи ПО часто обращаются с такими просьбами. Влияние таких изменений может быть просто огромно. Все инструментальные средства SCM поддерживают механизм для управления соответствующими изменениями. |
| Групповой синдром разработчика | Если для разработки проекта требуется более одного разработчика, то возникает проблема группы людей, работающих над одной базой продукта. В данном случае базой может быть план проведения испытаний, требования к спецификации или коду. Усилия тратятся впустую, несколько человек работают над одним и тем же файлом, а затем его сохраняют. Без контроля УК И УИ сохраняются только изменения, записанные последним, кто работает над этим файлом; все остальные изменения — теряются. Самое простое решение проблемы лежит в блокировании файла на время работы с ним одним из сотрудников, чтобы предотвратить одновременную работу над ним не скольких человек. |
| Множественность версий | Совершенствование базового продукта приводит к выпуску дополнительных версий с самыми последними изменениями. Несмотря на наличие последней модернизированной версии программного продукта, часть пользователей продолжает работать с более ранней версией. Продукт УК И УИ позволяет контролировать все версии. Если в программе обнаружены ошибки, то изменения необходимо сделать во всех версиях. Как только в продукте появляются новые свойства, они должны быть доступны для всех пользователей независимо от времени выпуска версии продукта. |
| Семейство программных продуктов | Поскольку программные продукты созданы для того, чтобы предлагать аналогичные функции с помощью неоднородных платформ аппаратного обеспечения, необходимо управлять как программным продуктом вообще, так и ПО на базе определенной аппаратной платформы. Если программный продукт работает с четырьмя версиями Windows, тремя версиями Unix, Red Hat Linux и FreeBSD, то руководство для пользователя должно быть аналогичным. Но для этих девяти платформ необходимы разные процессы установки ПО. Без применения о написать девять отдельных руководств пользователя по данному программному продукту. При наличии УК И УИ достаточно будет одного комплекта документации, куда будут входить все девять версий, которые будут отличаться лишь процедурой инсталляции программного продукта. |
| Изменение требований | Первый закон системотехники заключается в том, что независимо от этапа жизненного цикла системы, система/ПО будет изменяться, а желание изменить ее будет постоянно возникать на всем протяжении жизненного цикла продукта. Борьба с такими изменениями представляет собой сложную управленческую проблему. Наличие УК И УИ облегчает управление такими изменениями требований к продукту. УК И УИ позволяет легко идентифицировать наборы функциональных возможностей, которые объединяют требования, которым отвечает версия продукта. Этот набор функциональных возможностей отслеживается от разработки до поставки продукта. |
| Изменение графика работ | Поскольку технические требования изменяются, должен изменяться и график их выполнения. Составление календарного плана с учетом набора функциональных возможностей для версии позволяет менеджерам проектов более точно распределять силы, необходимые для выпуска следующей версии программного продукта. Наличие УК И УИ дает возможность на основе статистических данных сравнивать эффективность работы при подготовке новых версий. Статистические данные помогают оценить развитие событий типа а что, если, которые происходят в результате успешного внедрения программного продукта среди новых потребителей или предоставлении выполненных по заказу продуктов другим клиентам. |
| Изменения ПО | Ни один разработчик не позволяет себе, однажды написав программу, полностью о ней забыть. Разрабатываемое ПО изменяется не только при изменении технических требований и календарных планов, но и в ответ на изменения в других элементах. ПО не является догмой. В этом и заключается его ценность. Программный продукт можно изменять, поэтому его и изменяют. Системы УК И УИ отслеживают эти изменения, а если внесено неверное изменение, то всегда можно посмотреть предыдущую рабочую версию. Только одна эта функция УК И УИ экономит огромное количество времени, поскольку разработчики проверяют конкретные задачи, которые не работают в среде программного продукта, и могут быстро перейти к рабочей версии. |
| Изменения штата | Во всех организациях сотрудники продвигаются по служебной лестнице, переходят на другую работу или увольняются. Если это происходит в разгар работы по разработке ПО, то с уходом специалиста теряются не только технологические знания. Теряются также практические знания по разработке продуктов, на овладение которыми ушло много времени. Новые сотрудники, даже зная технологию, не смогут заниматься разработкой продукта без задокументированного процесса УК И УИ. Таким образом, УК И УИ является точкой отсчета и базой данных об истории разработки проекта. Благодаря УК И УИ новый сотрудник может узнать, как идет процесс разработки в организации и что нового в проекте на конкретную дату. |
| Изменения документации система/пользователь | Ни один разработчик не имеет права что-то пропустить в технологии или инструментальном средстве. Все разработчики продукта используют систему микропрограмм аппаратных средств, операционные системы, инструментальные средства и документацию, которые не находятся под их контролем. При изменении основной операционной системы (например, следующей самой лучшей версии Windows) УК И УИ отслеживает все элементы конфигурации, компоненты и подкомпоненты, на которые может оказать влияние это изменение. Каждое изменение анализируют отдельно, что позволяет правильно распределить силы, необходимые для реагирования на это изменение. Можно составить календарный план реагирования на ситуацию, которая выходит за рамки контроля этой организации. |