**Экстремальное программирование**

Extreme Programming (**XP**) - это дисциплина разработки ПО и ведения бизнеса в области создания программных продуктов, которая фокусирует усилия программистов и бизнесменов на общих вполне достижимых целях.

Команды, использующие **XP** производят качественное ПО с весьма большой скоростью. Методики XP основаны на человеческом творчестве и принятии того, что человек является существом неустойчивым и подверженным ошибкам.

**XP** формирует 2 набора обещаний:

* **Обещание программистам**: программистам XP обещает, что каждый из них будет работать над решением действительно важных задач каждый рабочий день. Каждый из программистов никогда не окажется в затруднительном положении в одиночку. Каждый из программистов будет способен сделать все от него зависимое, чтобы сделать разрабатываемую систему успешной. Каждый из программистов будет способен принять решение именно в той области, в которой он компетентен и если он в некоторых областях недостаточно компетентен, он не будет участвовать в принятии решений.
* **Обещание заказчикам**: заказчикам и менеджерам XP обещает, что они получат максимально возможную отдачу от каждой недели работы над проектом. Каждые несколько недель будут способны увидеть прогресс в достижении заданных ими целях. Они получат возможность изменить направление развития проекта в самой середине разработки, не опасаясь при этом дополнительных экстраординарных затрат.

XP обещает снизить связанный с разработкой проекта риск, улучшить реакцию на изменения бизнеса, улучшить производительность работы над проектом и сделать процесс разработки ПО более приятным. XP - это упрощенный, эффективный, гибкий, предсказуемый, научно-обоснованный и весьма приятный способ разработки ПО, предусматривающий низкий уровень риска.

В основе XP в первую очередь лежат не конкретный методики, как это принято считать, а четыре базовых принципа:

1. Общение
2. Простота
3. Обратная связь
4. Храбрость

XP предлагает готовое решение: делайте все максимально просто, держите заказчика при себе или сами держитесь при заказчике, позвольте ему активно следить за процессом разработки, приветствуйте изменения и успех практически обеспечен.

**Методики XP**

* **Игра в планирование**: при разработке ПО редко можно сказать, что окончательный вид системы заранее известен еще в самом начале разработки. Многое меняется в процессе самой разработки. В XP это неотъемлемая часть разработки и то, что планы могут поменяться, учитывается с самого начала. Игра в планирование помогает прогнозировать ситуацию и подготовиться к ней морально. Суть методики в следующем: ключевой фигурой игры является заказчик: именно он сообщает о необходимости той или иной функциональности, а программисты дают ориентировочную оценку каждой функциональности, при этом обе стороны понимают, что у них единая цель и они должны быть солидарны друг с другом, так как в случае победы выигрывает все, а в случае поражения все проигрывают. При этом каждый идут своим путем: заказчик выбирает наиболее важные задачи в соответствии с бюджетом, а программисты оценивают задачи в соответствии со своими возможностями по их реализации. В идеале игра в планирование с участием заказчика и программиста должна проводится каждые 3-6 недель до начала следующей итерации разработки.
* **Тестирование до начала разработки**: в XP в начале идет тест, а потом код. Соответственно, возникает вопрос, как же тестировать то, чего нет? – XP предлагает тестировать свои мысли: то есть, чего следует ожидать от будущего куска программы или функциональности. Это позволит понять, что требуется сделать программистам и проверить работоспособность кода сразу как он будет написан. Такой подход ускоряет разработку. Без знания того, что нужно сделать и требуемого объема работа экономит время.
* **Парное программирование**: периодический просмотр чужого кода положительно влияет на его качество. XP использует этот подход, в ходе разработки код просматривается постоянно при помощи парного программирования: два программиста работают за одним компьютером, пользуясь общими клавиатурам и мышкой. Один пишет, другой поправляет, следит за ошибками. Потом они меняются местами и т.д. Исследования показали, что затраты на разработку не увеличиваются вдвое, а за счет экономии времен остаются приблизительно на том же уровне.
* **Постоянная переработка кода**:добавление новой функциональности и разрастания кода усложняют разработку, выявление ошибок и внесение изменений. XP рекомендует компенсировать добавление функциональности усовершенствованием кода. Это и есть переработка кода или рефакторинг. Переработка кода - это мощный инструмент
* обеспечения качества программы.
* **Простота разработки**: XP рекомендует делать все максимально просто. Простую программу легче поддерживать, в нее легче вносить изменения, она менее подвержена ошибкам.
* **Коллективное владение кодом**: традиционно программный продукт поделен на сферы влияния между несколькими разработчиками. В такой схеме каждый разработчик - это специалист узкого профиля. Такой подход приводит к проблемам если один из разработчиков идет в отпуск, заболеет, уволиться. XP предлагает компенсировать возможные выпадающие звенья используя принцип коллективного владения кодом. Каждым участком кода как минимум должны владеть два программиста, и любой участник команды может внести изменения в любой фрагмент кода
* **Продолжающаяся интеграция**: часто разработчики сталкиваются с проблемами, когда новая функциональность оказывается несовместима с остальным проектом. XP учит, что эффективный способ решения данной проблемы заключается в том, чтобы в первую очередь реализовывать сложный или незнакомый ранее функционал, при этом интегрировать в программу работоспособный участок кода как можно раньше. Такой способ может побороть или даже предотвратить несовместимость на ранней стадии проекта.
* **Заказчик на рабочей площадке**: основная проблема разработки ПО – это недостаток знаний программистов в разрабатываемой предметной области. Поэтому XP рекомендует участие заказчика в процессе разработки. Заказчики бывают разные. Если привлечь заказчика или его представителя не удается, то иногда оказывается целесообразным временный наём специалиста в разрабатываемой области. Такой шаг сократит неясности в работе, повысит скорость разработки и приблизит проект к тому, что желает получить заказчик.
* **Быстрый выпуск версий**: ни один аналитик не поймет потребностей заказчика лучше, чем сам заказчик, попользовавшись готовым продуктом некоторое время. Эта особенность легла в основу этой методики: дайте заказчику определиться со своими потребностями раньше, выпустив версию быстрее. У такого подхода есть ограничения: пользователи не всегда готовы к быстрым переменам, выпускаемый продукт должен быть всецело протестирован.
* **40-часовая рабочая неделя**: человек, особенно если он программист, ради дела способе на многое: задержаться на работе, выйти на работу в выходные, отказаться от отпуска и т.д. XP категорически против такого самопожертвования и нарушения принятых норм трудового права. Это продиктовано не только из соображений гуманности и законности, а в первую очередь необходимостью повышения эффективности работы и строгой организации. Ведь XP – это коллективная игра, рассчитанная не на отдельных индивидуумов, а на всю группу целиком, а в группе немаловажную роль играет синхронизация биоритмов ее участников. Но самое главное человеку, чтобы сохранить здоровье и работоспособность необходим полноценный отдых. 8-часовой рабочий день и 5-дневная рабочая неделя установлены именно из соображений максимальной продуктивности. С медицинской точки зрения задержки на работе ведут к постоянной усталости, раздражительности и снижению мозговой деятельности.
* **Стандарты кодирования**: XP в качестве обязательного условия успешности работы над проектом выдвигает требования применения стандартов кодирования. Единообразный код более понятен другим разработчикам и задумываться при внесении изменений в чужой код относительно форматирования не придется.
* **Метафора системы**: первое представление о системе может быть получено посредством метафоры или сравнении с существующими аналогами. Человеческое мышление построено на образах: каждая слово вызывает в памяти ассоциацию с характерным образом. XP учитывает этот факт и рекомендует для понимая системы использовать сравнения наиболее похожее раннее известного продукта или предмета с разрабатываемым. Всегда легче запомнить мелкие отличия, чем строить в уме всю систему по кусочкам.

*Заключение*

Основное преимущество XP – это прогнозируемость и сведения к минимуму затрат на разработку, предоставления заказчику того продукта, который он желает получить.