

Прототипы как средство уменьшения риска

Предмет: Управление требованиями и проектирование ПО

Прототипы ПО могут быть действующими моделями или статичными образцами; детализированными изображениями экрана или его эскизами; наглядной демонстрацией или примерами реальной функциональности; имитацией или подражанием.

Прототип ПО — это частичная или возможная реализация предлагаемого нового продукта. Прототипы позволяют решать три основные задачи:

- Прояснение и завершение процесса формулировки требований
- Исследование альтернативных решений
- Создание конечного продукта

Горизонтальный прототип (поведенческий прототип) — в нем не реализуются все слои архитектуры и нюансы системы, но воплощаются некоторые особенности интерфейса пользователя.

Вертикальный прототип (структурный прототип, проверка концепции) — воплощает срез функциональности приложения от интерфейса пользователя до сервисных функций.

Одноразовый прототип (исследовательский прототип) — создается, чтобы ответить на вопросы, разрешить неясности и улучшить требования к ПО.

Эволюционный прототип — прочный архитектурный «фундамент» для постепенного создания окончательного продукта — по мере прояснения требований.



Последовательность действий от вариантов использования к дизайну интерфейса пользователя через одноразовый прототип

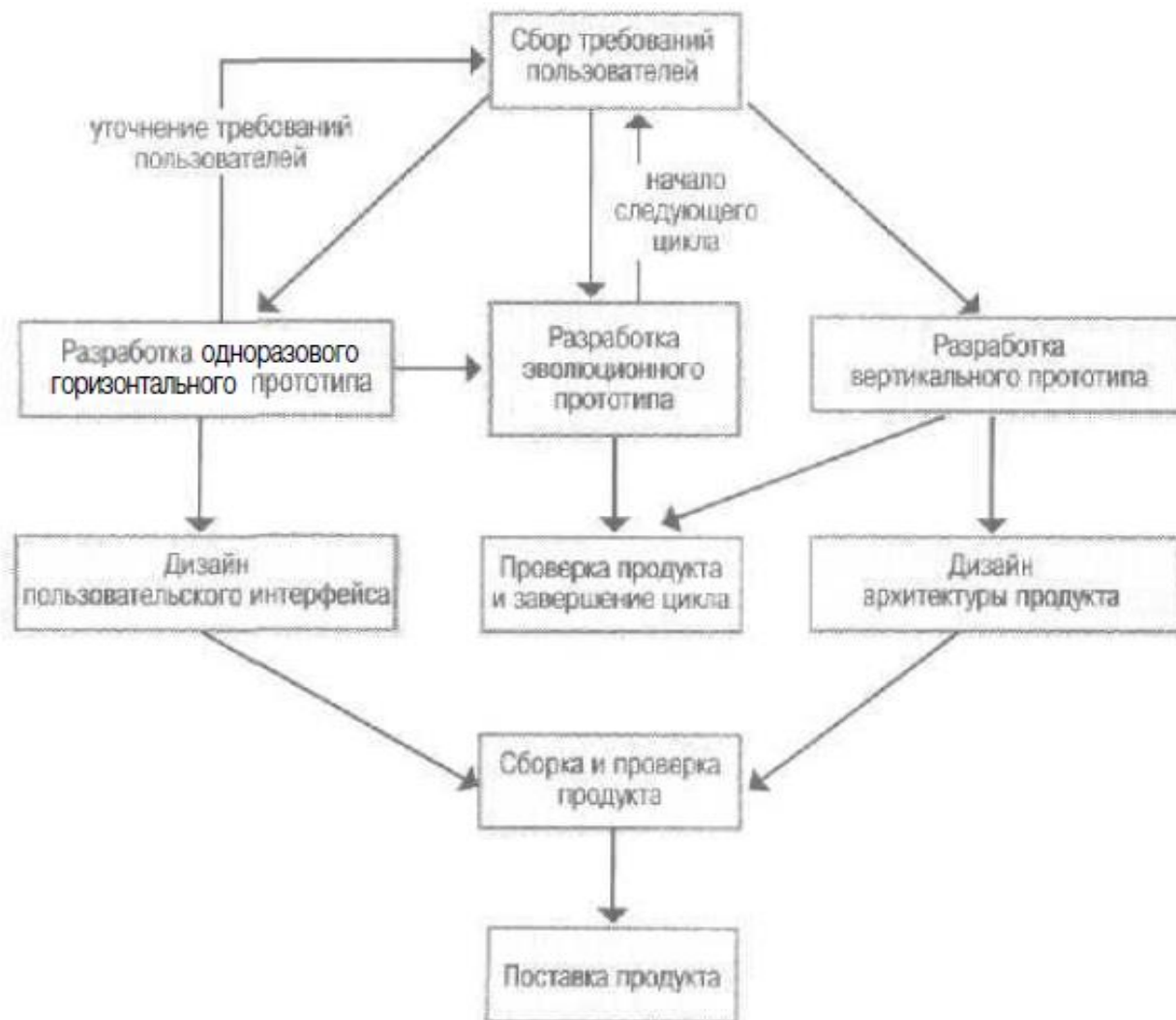


Рис. 13-2. Несколько возможных способов использования прототипов в процессе разработки ПО

Таблица 13-1. Стандартные способы применения прототипов ПО

	Одноразовые	Эволюционные
Горизонтальные	<ul style="list-style-type: none"> ■ Прояснение и уточнение примеров использования и функциональных требований ■ Выявление пропущенных функций ■ Исследование возможных вариантов интерфейса пользователя 	<ul style="list-style-type: none"> ■ Реализация базовых вариантов использования ■ Реализация дополнительных вариантов использования по приоритетам ■ Реализация и доработка Web-сайтов ■ Адаптация системы к быстро меняющимся требованиям бизнеса
Вертикальные	<ul style="list-style-type: none"> ■ Демонстрация технической осуществимости 	<ul style="list-style-type: none"> ■ Реализация и наращивание ключевой клиент-серверной функциональности и уровней коммуникации ■ Реализация и оптимизация основных алгоритмов ■ Тестирование и настройка производительности

Бумажный (низкокачественный прототип) — это дешевый, быстрый и низкотехнологичный способ выяснить, как может выглядеть некий фрагмент системы.

Какими бы совершенными ни были ваши инструменты прототипирования, наброски экранов на бумаге делать все равно быстрее. Бумажное прототипирование ускоряет каждый цикл, а это — ключевой фактор успеха при разработке требований. Это отличный метод уточнения требований для проектирования детализированных интерфейсов пользователей, создания эволюционных прототипов или традиционного проектирования и конструирования. Он также помогает команде лучше управлять ожиданиями клиентов.

Факторы успеха прототипирования

- Включите задачи прототипирования в план проекта. Составьте график распределения затрат времени и средств на разработку, оценку и модификацию прототипов;
- Сформулируйте цель каждого прототипа до начала его создания
- Планируйте создание нескольких прототипов. В большинстве случаев вам не удастся создать то, что нужно, с первой попытки (в чем, собственно, и состоит весь смысл прототипирования);
- Одноразовые прототипы создавайте так быстро и дешево, как возможно. Вкладывайте минимум усилий в подготовку прототипов, которые помогут отвечать на вопросы или разрешать неясности в требованиях. Не доводите до совершенства одноразовые прототипы;
- Не встраивайте в одноразовый прототип подробную проверку входных данных, методы безопасного программирования, процедур обработок ошибок или документации кода;
- Не прототипируйте элементы, которые уже понимаете, кроме случаев исследования альтернатив дизайна;
- Используйте правдоподобные данные в экранных формах и сообщениях прототипа. Пользователи будут отвлекаться и не смогут воспринять прототип как модель, описывающую внешний вид и поведение реальной системы;
- Не ожидайте, что прототип полностью заменит спецификацию требований к ПО. Большая часть скрытой функциональности лишь подразумевается в прототипе и должна быть задокументировано в спецификации требований к ПО, чтобы ее удалось реализовать полно, точно и прозрачно. Видимая часть приложения — это пресловутая вершина айсберга. Изображения экранов не дают деталей определения полей данных, критериев проверки, взаимосвязей между полями (таких, как элементы управления пользовательского интерфейса, появляющихся, только если пользователь совершает определенные действия с другими элементами управления), обработки исключений, бизнес-правил и др.