

Манифест Реактивных Систем

Published on September 16 2014. (v2.0)

Организации, работающие в совершенно разных отраслях, независимо друг от друга находят для себя похожие шаблоны создания программного обеспечения. Основанные на них системы оказываются более надежными, устойчивыми, гибкими и соответствующими современным требованиям.

В последние годы требования к приложениям кардинально изменились. Всего несколько лет назад большим считалось приложение, состоящее из десятков серверов, тогда время ответа измерялось секундами, простои при техническом обслуживании - часами, а данные для обработки - гигабайтами. Сегодня приложения устанавливаются на всем, от мобильных устройств до облачных кластеров, насчитывающих тысячи многоядерных процессоров. Пользователи привыкли к миллисекундным задержкам и стопроцентной доступности, а данные тем временем измеряются петабайтами. Программные архитектуры вчерашнего дня не способны удовлетворить требованиям дня сегодняшнего.

Мы считаем, что для проектирования приложений необходим четкий и понятный подход, и что все его отдельные аспекты уже сформулированы: мы хотим, чтобы система была доступной, устойчивой, гибкой и основанной на обмене сообщениями. Мы называем такие системы Реактивными.

Приложения, написанные в соответствии с принципами Реактивных Систем, отличаются повышенной гибкостью, [масштабируемостью](#) и слабой связанностью. Благодаря этому их проще разрабатывать и модифицировать. Они более устойчивы к [отказам](#) и корректно обрабатывают исключительные ситуации, избегая катастрофических последствий. Реактивные системы характеризуются высокой доступностью, обеспечивая [пользователям](#) эффективную и интерактивную обратную связь.

Реактивные Системы:

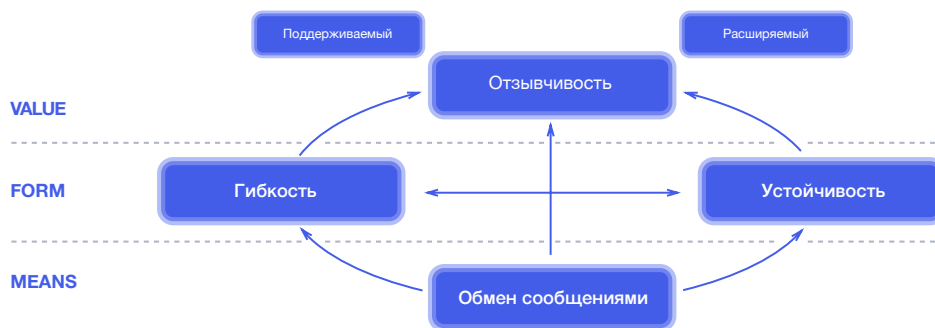
Доступные: Система отвечает своевременно, если это вообще возможно. Доступность является краеугольным камнем удобного и полезного приложения, но, помимо этого, она позволяет быстро обнаруживать проблемы и эффективно их устранять. Доступные системы ориентированы на обеспечение быстрого и согласованного времени отклика, устанавливая надежные верхние границы, чтобы обеспечить стабильное качество обслуживания. Такое предсказуемое поведение, в свою очередь, упрощает обработку ошибок, повышает уверенность конечного пользователя в работоспособности и способствует дальнейшему взаимодействию с системой.

Устойчивые: Система остается доступной даже в случае [отказов](#). Это относится не только к высокодоступным, критически важным приложениям — без устойчивости любая система при сбое теряет доступность. Устойчивость достигается за счет [репликации](#), сдерживания, [изоляции](#) и [делегирования](#).

Эффект от отказов удерживается внутри [компонентов](#), изолируя их друг от друга, что позволяет им выходить из строя и восстанавливаться, не нарушая работу системы в целом. Восстановление каждого компонента делегируется другому (внешнему) модулю, а высокая доступность обеспечивается за счет репликации там, где это необходимо. Клиент компонента не отвечает за обработку его сбоев.

Гибкие: Система остается доступной под разными нагрузками. Реактивные Системы способны реагировать на колебания в скорости входящих потоков, увеличивая или уменьшая количество выделенных на их обслуживание [ресурсов](#). Для этого архитектура не должна допускать наличия централизованных узких мест или конкуренции за ресурсы, что позволяет сегментировать или реплицировать компоненты, распределяя между ними входные данные. Реактивные Системы поддерживают предсказывающие и Реактивные алгоритмы масштабирования, позволяя делать измерения производительности в режиме реального времени. [Гибкость](#) достигается применением экономически эффективных аппаратных и программных платформ.

Основаны на обмене сообщениями: Реактивные системы используют [асинхронный обмен сообщениями](#), чтобы установить границы между компонентами и обеспечить слабую связанность, изоляцию и [прозрачность размещения](#). Эти границы также позволяют преобразовывать и передавать информацию о [сбое](#) в виде сообщений. Открытый обмен сообщениями делает возможными регулирование нагрузки, гибкость и управление потоком, для чего в системе создаются и отслеживаются очереди сообщений и в случае необходимости используется [обратное давление](#). Прозрачность размещения при взаимодействии на основе сообщений позволяет применять к механизму обработки ошибок одни и те же ограничения и семантику как в пределах одного компьютера, так и в масштабах целого кластера. Благодаря [неблокирующему](#) взаимодействию принимающая сторона потребляет [ресурсы](#) только при активной работе, что позволяет снизить накладные расходы.



Большие системы состоят из подсистем, имеющих те же свойства и, следовательно, зависят от их реактивных характеристик. Это означает, что принципы Реактивных Систем применяются на всех уровнях, что позволяет компоновать их между собой. Архитектура, основанная на этих принципах, реализована в самых масштабных приложениях в мире, ежедневно обслуживающих миллиарды пользователей. Настало время сознательно применять эти принципы проектирования, а не заново открывать их каждый раз.