

Характеристики

Требование	Хар-ка	Сравнение систем				
Мы ожидаем 1к заявок в день от рандомных котов, также, судя по отзывам, наши конкуренты могут попытаться нас заддосить в этом месте. Общая нагрузка на систему не будет превышать 10 заказов в день и 100 клиентов. Воркеров будет около 20 котов.	Availability Scalability Elasticity	elasticity	★	★	★★	★★★★★★
		scalability	★	★	★★★	★★★★★★
Как минимум одна часть функционала будет высоконагруженной. В микросервисах удобнее разделить функционал по уровню нагрузки						
Бизнесу необходим низкий TTM (Time To Market), чтобы конкурировать на рынке.	Deployability Testability	deployability	★	★★	★★★★	★★★★
		testability	★★	★★	★★★★	★★★★
Для бизнеса критично проверять новые гипотезы по отсеvu котов и изменять уже существующие с максимальной скоростью и надёжностью. Расчёт стоимости задаётся по секретной формуле, которую пока никто не знает, поэтому для первой версии хватит задать константное значение в 100 условных кото-единиц за любую услугу. Исследователи MCF пока думают о том, как должна работать инновационная система матчинга (тм). Так как без матчинга нельзя, процесс должен рандомно связывать воркера и клиента, без специфики.	Agility Modifiability Maintainability	modifiability	★★	★★★	★★	★★★★★★
		agility	★	★★	★★★★	★★★★★★
		maintainability	★★	★★★	★★	★★★★★★
При запросе услуги клиенты будут видеть форму, в которой можно описать, что	Usability Consistency Simplicity	simplicity	★★★★★★	★★★★★★	★★★	★
		Нужны понятный интерфейс – для заведения услуг, для проведения их, для проверки качества выполнения и просмотра отчетов; консистентность данных, чтобы данные об услугах были одинаковы для				

<p>нужно сделать, выбрать тип услуги, увидеть примерную стоимость.</p> <p>Когда воркер придёт в нужное время на заказ — ему необходимо отметить в приложении (прислать фотографию и нажать кнопку «приступил»). В конце работы необходимо прислать фотографию бумажного отчёта с подписью клиента, что всё выполнено и претензий нет.</p> <p>Сотрудники отдела расходников сами собирают заказы на свой вкус, нам хватит только уведомить сотрудников о новом заказе и предоставить кнопку «выдано» под каждый заказ.</p> <p>Клиент и воркер должны видеть список всего, за что у них списались или начислились деньги.</p> <p>Необходимо показывать, сколько клиент заплатит денег в конце текущей недели, а также список всех прошлых инвойсов и статус их оплат. Для воркера нужно показывать, сколько он заработал, штрафы и также инвойсы с их статусом.</p> <p>Все отменённые или провальные заказы автоматически попадают в отдел по изучению качества работы.</p> <p>После выполнения или отмены или провала</p>		<p>всех участников, чтобы деньги списывались и зачислялись на одинаковые суммы и т.д.</p> <p>В микросервисах за этим всем следить сложнее, так как данные гуляют по разным сервисам, возможность дублирования или потери части данных выше, чем в других архитектурах.</p>
--	--	--

заказа система автоматически рассчитывает, кому и сколько денег надо отдать (банк делится между победителями).		
Деньги на данный момент не критичны, harpu sat box готовы потратить столько, сколько потребуется.	Cost	<div><div>cost</div><div><div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div></div><div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div></div><div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div><div>★</div></div><div><div>★</div></div></div><div>Не переживаем по поводу стоимости возможного решения</div></div>
<p>Чтобы печенье попало к клиенту, необходимо отправить информацию о клиенте подрядчику, и через 10 минут печенье приедет на склад, после чего сотрудники положат его в нужный набор расходников.</p> <p>Для этого выставляется инвойс, и дальше происходит процесс зачисления средств через золотую шляпу (уникальный сервис для перевода денег, никак не связанный с «золотой короной»).</p>	Configurability Extensibility Reuse	<div><div>configurability</div><div><div>★</div></div><div><div>★</div></div><div><div>★</div><div>★</div></div><div><div>★</div><div>★</div><div>★</div></div></div> <div>В микросервисной архитектуре удобно выделять отдельные сервисы под отдельную интеграцию с внешней системой, чтобы не разводить зоопарк из разных технологий в одном сервисе.</div>