15

O‘N BESHINCHI MA’RUZA

Korxona ilovalarini dasturlash · 15-mavzu

Dasturiy ta’minot hayotiy sikli va tendensiyalar

Hayotiy sikl · hujjatlar · qo‘llab-quvvatlash · o‘zgarishlarni boshqarish · tendensiyalar · himoya

SDLC | SemVer | ADR | Feature flag | AI

ISINISH

OmborPro himoyadan keyin 5 yil ishlasa, eng ko‘p vaqt va pul qaysi bosqichga ketadi?

Ishlab chiqishgami yoki undan keyingi hayotigami? Taxmin qiling va asoslang.

1

Laravel ning siz ishlatgan versiyasi uchun xavfsizlik tuzatishlari tugasa, ilova bilan nima qilish kerak?

2

Jamoaga yangi dasturchi qo‘shilsa, loyihani qaysi hujjatdan boshlab o‘rganadi?

3

Buxgalter kechqurun “hisob-faktura PDF noto‘g‘ri” deb yozdi — kim, qancha vaqtda va qanday javob beradi?

O‘QUV MAQSADLARI

Bu ma’ruza oxirida siz quyidagilarni bajara olasiz…

1. **Tavsiflash** — dasturiy ta’minot hayotiy sikli bosqichlarini va ISO/IEC 14764 bo‘yicha texnik xizmat turlarini
2. **Yozish** — README, ADR, OpenAPI (Scribe) va foydalanuvchi qo‘llanmasidan iborat hujjatlar to‘plamini
3. **Qo‘llash** — SLA, P1–P4 ustuvorliklar, runbook va aybsiz postmortem asosidagi qo‘llab-quvvatlash jarayonini
4. **Boshqarish** — o‘zgarishlarni SemVer, Keep a Changelog, Conventional Commits va git teglar yordamida
5. **Qo‘llash** — Laravel Pennant feature flaglari orqali bosqichma-bosqich va qaytariladigan relizni
6. **Farqlash** — modulli monolit, mikroservislar va low-code yondashuvlarining qo‘llanish sohalarini
7. **Baholash** — AI yordamchilarning foyda va xatarlarini moliya-bank ma’lumotlari kontekstida
8. **Tayyorlash** — yakuniy loyiha himoyasi uchun reliz, demo ssenariysi va zaxira rejani

01

Hayotiy sikl va texnik xizmat

SDLC bosqichlari · egalik qiymati · ISO/IEC 14764 bo‘yicha xizmat turlari

SDLC | ISO/IEC 14764 | Texnik xizmat | TCO

SDLC

Hayotiy sikl — himoyadan keyin ham davom etadi

SDLC — talablardan to ishdan chiqarishgacha bo‘lgan bosqichlar. Joriy etish — oxiri emas, eng uzun bosqichning boshlanishi.

Ishlab chiqish

Talablar, loyihalash, kod, testlar — odatda bir necha oy.

Qo‘llab-quvvatlash

Xatolar, qonun va integratsiya o‘zgarishlari, yangilanishlar — yillar.

TCO

Umumiy egalik qiymati: litsenziya, server, ishlab chiqish va xizmat xarajatlari yig‘indisi.

Ishdan chiqarish

Ma’lumotlarni ko‘chirish, arxivlash va tizimni xavfsiz o‘chirish.

Turli manbalarga ko‘ra TCO ning 60–80 % gacha qismi joriy etilgandan keyingi davrga to‘g‘ri keladi.

BOSQICHLAR

SDLC bosqichlari va OmborPro dagi natijalar

Bosqich	Asosiy natija	OmborPro da (mavzu)
Talablar	User stories, backlog	Loyiha xartiyasi (1)
Loyihalash	Arxitektura, ER-sxema, ADR	Qatlamlar, modellar (2–3)
Ishlab chiqish	Kod, PR, code review	Servislar, UI, API (4–11)
Testlash	Pest testlari, qamrov	Testlash (12)
Joriy etish	Docker, CI/CD, VPS	DevOps (14)
Qo‘llab-quvvatlash	SLA, monitoring, relizlar	Kuzatuvchanlik (13), bugun (15)
Ishdan chiqarish	Ma’lumot ko‘chirish, arxiv	Yangi tizimga migratsiya

Agile da bu bosqichlar har sprintda qisqa sikl bo‘lib takrorlanadi.

ISO/IEC 14764

Texnik xizmatning to‘rt turi

Har bir o‘zgarish so‘rovi shu to‘rt turdan biriga kiradi — bu rejalashtirish va byudjet uchun asos.

1 Tuzatuvchi (corrective)

Topilgan xatoni tuzatish: chegirmada QQS 1 so‘mga noto‘g‘ri yaxlitlanadi.

2 Moslashtiruvchi (adaptive)

Muhit o‘zgarishiga moslash: cbu.uz API formati, yangi Laravel versiyasi.

3 Takomillashtiruvchi (perfective)

Yangi imkoniyat yoki tezlik: Excel eksport, hisobotni keshlash.

4 Oldini oluvchi (preventive)

Yashirin muammoni oldindan bartaraf etish: refaktoring, testlar, paket yangilash.

Amalda jamoalar sprint vaqtining bir qismini (masalan, 20 %) oldini oluvchi ishlarga ajratadi.

▶ INTERAKTIV

Bu qaysi xizmat turi?

ISO/IEC 14764 bo‘yicha o‘zgarish turini aniqlang. (Interaktiv slaydlarda bosib bajariladi.)

1 **Payme webhook ikki marta kelganda to‘lov ikki marta yozilyapti. Tuzatish qaysi tur?**

A) Moslashtiruvchi
B) Tuzatuvchi
C) Takomillashtiruvchi
D) Oldini oluvchi

2 **Laravel 11 ning xavfsizlik tuzatishlari muddati tugadi, Laravel 12 ga o‘tamiz. Qaysi tur?**

A) Tuzatuvchi
B) Takomillashtiruvchi
C) Moslashtiruvchi
D) Xizmat emas

3 `OrderService` **dagi 300 qatorli metod kichik metodlarga ajratildi, xatti-harakat o‘zgarmadi.**

A) Oldini oluvchi
B) Tuzatuvchi
C) Moslashtiruvchi
D) Takomillashtiruvchi

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

1-bo‘lim yakuni: Hayotiy sikl va texnik xizmat

1

Nima uchun TCO ning katta qismi joriy etilgandan keyingi davrga to‘g‘ri keladi?

Savol 1

2

ISO/IEC 14764 dagi to‘rt xizmat turini sanang.

Savol 2

3

Markaziy bank API si javob formatini o‘zgartirsa, bu qaysi xizmat turi?

Savol 3

02

Hujjatlashtirish

README · ADR · OpenAPI (Scribe) · foydalanuvchi qo‘llanmasi · docs-as-code

README | ADR | OpenAPI | Docs-as-code

Kim uchun, nima va qayerda

Hujjat	Kim uchun	Tarkibi	Qayerda
README	Dasturchi	Maqsad, ishga tushirish, testlar, `.env`	`README.md`
ADR	Jamoa, kelajakdagi dasturchi	Kontekst, qaror, oqibatlar	`docs/adr/`
OpenAPI	Integratorlar (1C, mobil)	Endpointlar, sxemalar, xatolar	Scribe → `/docs`
Qo‘llanma	Admin, menejer, omborchi, buxgalter	Jarayonlar, skrinshotlar	`docs/user-guide/`
CHANGELOG	Hamma	Versiyalar va o‘zgarishlar	`CHANGELOG.md`
Runbook	Navbatchi dasturchi	Hodisalarda tayyor buyruqlar	`docs/runbook.md`

Docs-as-code: hamma hujjat Markdown da, repozitoriyda, PR orqali o‘zgaradi.

ADR

Arxitektura qarori yozuvi (Nygard formati)

```
# ADR-0004: Modulli monolit arxitekturasi

Holat: Qabul qilingan (2026-11-20)

## Kontekst
Jamoa 4 kishi, bitta PostgreSQL, 15 haftalik muddat.
Rezerv va hisob-faktura bitta tranzaksiyada bo'lishi shart.

## Qaror
Bitta Laravel ilovasi; Sales, Inventory, Billing modullari
bir-biriga faqat servis interfeyslari orqali murojaat qiladi.

## Oqibatlar
+ bitta deploy, ACID tranzaksiyalar, oddiy CI
- modul chegaralarini code review da nazorat qilish kerak
- Billing ni ajratish kerak bo'lsa — yangi ADR yoziladi
```

▶ NATIJA

`docs/adr/0004-modulli-monolit.md` — PR orqali qabul qilinadi
Keyin o‘zgarmaydi; yangi qaror → ADR-0007 (supersedes 0004)

Holat

proposed → accepted → deprecated / superseded

Kontekst

qaror qabul qilingan paytdagi cheklovlar — keyin eng qimmatli qism

Oqibatlar

faqat afzalliklar emas, salbiy tomonlar ham yoziladi

Raqam

`0004` — ketma-ket, qayta ishlatilmaydi

OPENAPI

Scribe: hujjat koddan generatsiya qilinadi

```php
/**
 * Mahsulotlar ro'yxati
 *
 * Faol mahsulotlarni sahifalab qaytaradi.
 *
 * @group Mahsulotlar
 * @authenticated
 * @queryParam category_id integer Kategoriya filtri. Example: 3
 * @queryParam per_page integer Sahifa hajmi. Example: 20
 */
public function index(Request $request): AnonymousResourceCollection
{
    $products = Product::query()
        ->where('is_active', true)
        ->paginate($request->integer('per_page', 20));

    return ProductResource::collection($products);
}
```

▶ NATIJA

```
$ php artisan scribe:generate
→ HTML hujjat (type = laravel bo‘lsa: /docs)
→ OpenAPI spetsifikatsiyasi + Postman kolleksiyasi
```

@group

endpointlarni hujjatda bo‘limlarga ajratadi

@queryParam

nom, tur, izoh va `Example:` qiymati

@authenticated

endpoint token talab qilishini belgilaydi

FormRequest

body parametrlari `rules()` dan avtomatik olinadi

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

2-bo‘lim yakuni: Hujjatlashtirish

1

ADR ning beshta asosiy qismi qaysilar?

Savol 1

2

Qabul qilingan ADR dagi qaror o‘zgarsa, nima qilinadi?

Savol 2

3

Nima uchun OpenAPI hujjatini qo‘lda emas, Scribe bilan generatsiya qilish afzal?

Savol 3

03

Qo‘llab-quvvatlash va texnik qarz

SLI/SLO/SLA · P1–P4 · triage va runbook · aybsiz postmortem · bog‘liqliklarni yangilash

SLA | Incident | Postmortem | composer audit | Texnik qa

USTUVORLIK

P1–P4: javob va tiklash muddatlari (namuna)

Daraja	Mezon	Javob	Tiklash
P1 — kritik	Tizim ishlamaydi yoki to‘lovlar to‘xtagan	15 daqiqa	4 soat
P2 — yuqori	Muhim funksiya ko‘pchilik (≥ 50) uchun ishlamaydi	1 soat	24 soat
P3 — o‘rta	Kichik guruhga ta’sir, aylanma yo‘l bor	4 soat	72 soat
P4 — past	Kosmetik xato, taklif	1 ish kuni	Keyingi reliz

Raqamlar — namuna; aniq qiymatlar mijoz bilan SLA da kelishiladi.

SLA HISOBI

Ustuvorlik va tiklash muddatini hisoblash

```php
function priority(bool $down, bool $money, int $users): string
{
    return match (true) {
        $down || $money => 'P1',
        $users >= 50 => 'P2',
        $users >= 5 => 'P3',
        default => 'P4',
    };
}
$hours = ['P1' => 4, 'P2' => 24, 'P3' => 72, 'P4' => 240];
$tz = new DateTimeZone('Asia/Tashkent');
$opened = new DateTimeImmutable('2026-12-14 09:30', $tz);
$cases = [[false, true, 3], [false, false, 60], [false, false, 1]];
foreach ($cases as $c) {
    $p = priority(...$c);
    $due = $opened->modify("+{$hours[$p]} hours");
    echo $p, ' -> ', $due->format('d.m.Y H:i'), PHP_EOL;
}
```

▶ NATIJA

```
P1 -> 14.12.2026 13:30
P2 -> 15.12.2026 09:30
P4 -> 24.12.2026 09:30
```

`match (true)`

birinchi rost shart g‘olib — tartib muhim

`...$c`

massivni argumentlarga yoyish (spread)

`Asia/Tashkent`

muddat mijoz vaqtida hisoblanadi

`modify()`

`DateTimeImmutable` yangi obyekt qaytaradi, asl sana o‘zgarmaydi

HODISA JARAYONI

Signaldan postmortemgacha

1

Aniqlash

Sentry xatosi, `/up` monitoringi yoki foydalanuvchi murojaati

2

Triage

Takrorlanadimi, kimga ta’sir qiladi, ma’lumot buzilganmi → P1–P4

3

Yumshatish

Runbook: flagni o‘chirish, oldingi relizga qaytish, navbatni qayta ishga tushirish

4

Tuzatish

Test bilan PR, code review, patch reliz (`1.4.3`)

5

Postmortem

Aybsiz tahlil: xronologiya, asosiy sabab, chora-tadbirlar → backlog

Avval xizmatni tiklash, keyin sababni tuzatish: P1 da “chiroyli yechim” kutib o‘tirilmaydi.

Paketlarni muntazam tekshirish va yangilash

```
# 1. Eskirgan to'g'ridan-to'g'ri bog'liqliklar
composer outdated --direct

# 2. Ma'lum zaifliklar (CI da ham ishga tushiring)
composer audit

# 3. Minor/patch yangilash - composer.json chegarasida
composer update sentry/sentry-laravel --with-dependencies

# 4. Major yangilash - alohida branch, ADR va to'liq testlar
git switch -c upgrade/laravel-12
composer require laravel/framework:^12.0 -W
php artisan test
```

▶ NATIJA

```
$ composer audit
No security vulnerability advisories found.
```

`--direct`

faqat `composer.json` dagi paketlar, ichki bog‘liqliklarsiz

composer audit

paketlarni ma’lum zaifliklar bazasi bilan solishtiradi

`-W`

`--with-all-dependencies`: bog‘liq paketlarni ham yangilashga ruxsat

Major

Laravel upgrade guide ni o‘qish, alohida PR va to‘liq regressiya testi

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

3-bo‘lim yakuni: Qo‘llab-quvvatlash va texnik qarz

1

SLI, SLO va SLA ning farqi nimada?

Savol 1

2

Aybsiz (blameless) postmortemning maqsadi nima?

Savol 2

3

`composer outdated` va `composer audit` nimasi bilan farq qiladi?

Savol 3

MA’RUZADAGI MASHQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

1-topshiriq: Hodisa ustuvorligi va SLA muddati

Qo‘llab-quvvatlash xizmatiga murojaat keldi: `['down' => bool, 'payments' => bool, 'users' => int, 'opened' => 'Y-m-d H:i']`. Ustuvorlikni aniqlang: tizim ishlamasa (`down`) yoki to‘lovlar to‘xtagan bo‘lsa (`payments`) — `P1`; aks holda `users ≥ 50` — `P2`; `users ≥ 5` — `P3`; qolganlari — `P4`. Tiklash muddati: P1 — 4 soat, P2 — 24 soat, P3 — 72 soat, P4 — 240 soat (kalendar soatlari). `users` manfiy bo‘lsa yoki sana formati noto‘g‘ri bo‘lsa — `InvalidArgumentException`.

Imzo: `function incident_sla(array $i): array`

Qaytaradi: `['priority' => 'P1'|'P2'|'P3'|'P4', 'due' => 'Y-m-d H:i']` — kalitlar aynan shu tartibda.

MISOL · CHAQIRUV

```
incident_sla(['down' => true, 'payments' => false, 'users' => 0,
'opened' => '2026-12-14 09:30'])
```

MISOL · NATIJA

```
{"priority":"P1","due":"2026-12-14 13:30"}
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

04

O‘zgarishlarni boshqarish

SemVer · Keep a Changelog · Conventional Commits · migratsiyalar · Pennant feature flaglar

SemVer | CHANGELOG | Conventional Commits | Pennant

SEMVER

MAJOR.MINOR.PATCH — versiya bu va’da

Versiya raqami foydalanuvchiga yangilanish xavfsizmi yoki yo‘qligini aytadi.

MAJOR (2.0.0)

Orqaga mos emas: endpoint yoki maydon olib tashlandi, javob formati o‘zgardi.

MINOR (1.5.0)

Orqaga mos yangi imkoniyat: yangi endpoint, ixtiyoriy parametr.

PATCH (1.4.3)

Orqaga mos xato tuzatish: QQS yaxlitlash, validatsiya xabari.

Pre-release / build

`1.0.0-rc.1 < 1.0.0`; `+build.7` tartibga ta’sir qilmaydi.

`0.y.z` — boshlang‘ich bosqich: istalgan o‘zgarish bo‘lishi mumkin. Himoya relizi — `v1.0.0`.

VERSIYA HISOBI

Keyingi versiya va version_compare

```php
function bump(string $v, string $change): string
{
    [$major, $minor, $patch] = array_map('intval', explode('.', $v));
    return match ($change) {
        'breaking' => ($major + 1) . '.0.0',
        'feature' => "$major." . ($minor + 1) . '.0',
        'fix' => "$major.$minor." . ($patch + 1),
    };
}
echo bump('1.4.2', 'fix'), PHP_EOL;
echo bump('1.4.2', 'feature'), PHP_EOL;
echo bump('1.4.2', 'breaking'), PHP_EOL;

$list = ['1.10.0', '1.2.0', '1.0.0-rc.1', '1.0.0'];
usort($list, 'version_compare');
echo implode(' < ', $list), PHP_EOL;
```

▶ NATIJA

```
1.4.3
1.5.0
2.0.0
1.0.0-rc.1 < 1.0.0 < 1.2.0 < 1.10.0
```

`major + 1`

MINOR va PATCH nolga tushadi: `2.0.0`, `2.1.0` emas

`version_compare`

satr sifatida emas, raqamlar bo‘yicha: `1.10.0 > 1.2.0`

Ehtiyot bo‘ling

`version_compare` semver bilan to‘liq mos emas: `1.0.0-alpha.beta` ni `1.0.0-alpha.1` dan kichik deb hisoblaydi

Taqqoslash

`'1.10.0' > '1.9.0'` satr sifatida `false` — xato manbai

KEEP A CHANGELOG

CHANGELOG.md toifalari

Toifa	Nima yoziladi	OmborPro misoli
Added	Yangi imkoniyat	Debitorlik hisobotini Excel ga eksport
Changed	Mavjud xatti-harakat o‘zgardi	Buyurtma raqami formati `BY-2027-...`
Deprecated	Tez orada olib tashlanadi	`/api/v1/stocks` — v2 ga o‘ting
Removed	Olib tashlandi	Eski PDF shabloni
Fixed	Xato tuzatildi	Chegirmada QQS yaxlitlash
Security	Zaiflik yopildi	Eksportda ruxsat tekshiruvi qo‘shildi

Tepada `## [Unreleased]`; relizda u `## [1.1.0] - 2026-12-21` ga aylanadi va yangi bo‘sh `[Unreleased]` ochiladi.

LARAVEL PENNANT

Feature flag: deploy va release ni ajratish

```php
// app/Providers/AppServiceProvider.php - boot()
Feature::define('invoice-pdf-v2', function (User $user) {
    return match (true) {
        $user->hasRole('admin') => true,
        $user->hasRole('buxgalter') => Lottery::odds(1, 10),
        default => false,
    };
});

// Controller da - joriy (autentifikatsiyadan o'tgan) foydalanuvchi
if (Feature::active('invoice-pdf-v2')) {
    return $this->pdfV2->render($invoice);
}

// Navbatdagi job da - foydalanuvchi aniq ko'rsatiladi
$on = Feature::for($invoice->order->creator)->active('invoice-pdf-v2');
```

▶ NATIJA

```
admin       → true (har doim)
buxgalter   → ~10 % hollarda true; natija features jadvalida
saqlanadi
menejer     → false
O‘chirish: Feature::deactivateForEveryone('invoice-pdf-v2')
```

`Feature::define`

flag va uni hisoblash qoidasi (scope — `User`)

`Lottery::odds(1, 10)`

tasodifiy, lekin bir marta hisoblanib saqlanadi

`Feature::for($user)`

job va buyruqlarda joriy foydalanuvchi yo‘q — scope aniq beriladi

`@feature`

Blade da: `@feature('invoice-pdf-v2') ... @endfeature`

Qo‘llangan migratsiyani o‘zgartirish kerak bo‘lsa

Noto‘g‘ri: eski faylni tahrirlash

- `create_invoices_table` ga `currency` ustuni qo‘shib, faylni qayta saqlash
- Production da u allaqachon bajarilgan — `migrate` uni qayta ishga tushirmaydi
- Lokal, test va production sxemalari ajralib ketadi
- `migrate:fresh` bilan “tuzatish” — barcha ma’lumot yo‘qoladi

To‘g‘ri: yangi migratsiya

- `php artisan make:migration add_currency_to_invoices_table`
- Avval nullable yoki default qiymatli ustun (expand)
- Ma’lumotni ko‘chirish, kodni yangi ustunga o‘tkazish (migrate)
- Eski ustunni keyingi relizda olib tashlash (contract)

Migratsiya fayllari ham kod: ular PR da ko‘rib chiqiladi va CHANGELOG da qayd etiladi.

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

4-bo‘lim yakuni: O‘zgarishlarni boshqarish

1

`2.3.1` versiyasida orqaga mos bo‘lmagan o‘zgarish qilindi. Keyingi versiya qanday?

Savol 1

2

Keep a Changelog dagi oltita toifani sanang.

Savol 2

3

Navbatdagi job ichida `Feature::active()` nima uchun kutilgan natija bermasligi mumkin?

Savol 3

MA’RUZADAGI MASHQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

2-topshiriq: Conventional Commits bo‘yicha keyingi versiya

Joriy versiya `MAJOR.MINOR.PATCH` (faqat raqamlar, masalan `1.4.2`) va oxirgi relizdan keyingi commit xabarlari ro‘yxati berilgan. Sarlavha formati: `tur(soha)!: tavsif` — `(soha)` va `!` ixtiyoriy. Qoidalar: `!` bor yoki xabarda `BREAKING CHANGE:` uchrasa — MAJOR; `feat` — MINOR; `fix` yoki `perf` — PATCH; boshqa turlar (`docs`, `chore`, `refactor`...) va formatga mos kelmaydigan xabarlar versiyani o‘zgartirmaydi. Eng katta o‘zgarish hal qiladi. Versiya formati noto‘g‘ri bo‘lsa — `InvalidArgumentException`.

Imzo: `function next_version(string $current, array $commits): string`

Qaytaradi: Keyingi versiya satri, masalan `'1.5.0'`; hech qanday ta’sirli commit bo‘lmasa — joriy versiyaning o‘zi.

MISOL · CHAQIRUV	MISOL · NATIJA
`next_version('1.4.2', ['fix: QQS yaxlitlash', 'docs: README'])`	`"1.4.3"`

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

05

Tendensiyalar: mikroservislar, low-code, AI yordamchilar

Modulli monolit · mikroservislar narxi · low-code · AI va ma’lumotlar xavfsizligi

- Modulli monolit
- Mikroservislar
- Low-code
- AI yordamchilar

ARXITEKTURA

Modulli monolit va mikroservislar

Modulli monolit

- Bitta deploy, bitta CI, bitta monitoring
- ACID tranzaksiya: rezerv + hisob-faktura birga
- Modul chegaralari — interfeyslar va code review
- Kichik jamoa (3–10 kishi) uchun mos
- Kamchilik: butun ilova birga masshtablanadi

Mikroservislar

- Mustaqil deploy va masshtablash
- Har bir servisning o‘z bazasi va jamoasi
- Saga, retry, eventual consistency kerak
- Tarmoq, kuzatuvchanlik va versiyalash narxi yuqori
- Ko‘p jamoali yirik tashkilotlar uchun mos

“Monolith First”: avval modulli monolit, ehtiyoj isbotlangach — modulni servisga ajratish.

MODUL CHEGARASI

Sales moduli Inventory ga faqat interfeys orqali murojaat qiladi

```php
// Inventory modulining "eshigi"
interface StockReservation
{
    /** @param array<int, int> $items product_id => miqdor */
    public function reserve(int $warehouseId, array $items): void;

    public function release(int $orderId): void;
}

// StockService implements StockReservation - stocks jadvali
// faqat Inventory ichida o'qiladi va yoziladi

final class OrderService
{
    public function __construct(
        private StockReservation $stock,
    ) {}
}
```

▶ NATIJA

```
Sales  → StockReservation::reserve()   ✓
Sales  → DB::table('stocks')->update()  ✗ (code review da rad etiladi)
```

interface

modul ichki tuzilmasini yashiradi — keyin servisga ajratish oson

Konstruktor

Laravel konteyneri `StockService` ni bog‘laydi (`bind`)

Qoida

boshqa modul jadvaliga to‘g‘ridan-to‘g‘ri SQL yo‘q

Nazorat

code review yoki arxitektura testlari (masalan, Pest arch testlari)

LOW-CODE

Low-code qayerda o‘rinli, qayerda xavfli

1 O‘rinli

Ichki so‘rovnomalar, oddiy tasdiqlash jarayonlari, tezkor prototiplar.

2 Admin panellar

Filament kabi vositalar CRUD ni tez yasaydi — biznes-logika baribir servislarda.

3 Vendor lock-in

Platformadan chiqish qimmat: ma’lumot va jarayonlar yopiq formatda.

4 Murakkab qoidalar

Rezerv, QQS, holat mashinasi — vizual muharrirda boshqarish qiyin.

5 Soya IT

IT bilmagan ilovalar: ruxsatlar, audit, zaxira nusxa yo‘q.

Mezon: tizim qancha muhim va qancha uzoq yashaydi? Muhim va uzoq — odatiy dasturlash.

Tashqi AI servisiga yuborishdan oldin niqoblash

```php
function maskForAi(string $text): string
{
    $rules = [
        '/\b\d{9}\b/' => '[STIR]',
        '/\+998(?:[\s-]?\d){9}/' => '[TELEFON]',
        '/[\w.+-]+@[\w-]+\.[\w.]+/' => '[EMAIL]',
        '/\b\d{4}(?:\s?\d{4}){3}\b/' => '[KARTA]',
    ];
    return preg_replace(array_keys($rules), $rules, $text);
}
$msg = 'Mijoz STIR 301234567, tel +998 90 123-45-67, '
    . 'e-mail ali@example.uz, karta 8600 1234 5678 9012.';
echo maskForAi($msg), PHP_EOL;
```

▶ NATIJA

```
Mijoz STIR [STIR], tel [TELEFON], e-mail [EMAIL], karta [KARTA].
```

`preg_replace(massiv, massiv)`

qoidalar ketma-ket qo‘llanadi

`[STIR], [KARTA]`

model ma’noni tushunadi, lekin real qiymatni ko‘rmaydi

Cheklov

regex hamma holatni ushlamaydi — minimal ma’lumot yuborish asosiy himoya

Audit

yuborilgan so‘rov va javob `audit_logs` ga yoziladi

▶ INTERAKTIV

AI dan foydalanish: mumkinmi?

Har bir holatni moliya korxonasi siyosati nuqtai nazaridan baholang. (Interaktiv slaydlarda bosib bajariladi.)

AI bilan Pest test qoralamasini yozish, so‘ng code review

Mijozlar ro‘yxatini (STIR, telefon) tasdiqlanmagan chatbotga yuklash

README va qo‘llanma qoralamasini tayyorlash

AI tavsiya qilgan paketni tekshirmasdan `composer require` qilish

Murojaatni P1–P4 ga taxminiy toifalash, navbatchi tasdiqlaydi

Kredit limitini AI javobi asosida avtomatik tasdiqlash

Niqoblangan log parchasidan xato sababini izlash

`.env` faylini “tushuntirib ber” deb tashqi servisga yuborish

Mumkin (inson nazorati bilan)

Mumkin emas (yoki alohida ruxsatsiz)

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

5-bo‘lim yakuni: Tendensiyalar: mikroservislar, low-code, AI yordamchilar

1

Mikroservislarning modulli monolitga nisbatan uchta qo‘shimcha xarajatini ayting.

Savol 1

2

Low-code platformaning asosiy xatari nima?

Savol 2

3

Mijoz ma’lumotlarini tashqi AI servisiga yuborishdan oldin nimalar tekshiriladi?

Savol 3

MA’RUZADAGI MASHQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

3-topshiriq: AI servisiga yuborishdan oldin niqoblash

Matnni tashqi AI servisiga yuborishdan oldin shaxsga doir ma’lumotlar niqoblanadi. Qoidalar shu tartibda qo‘llanadi: (1) e-mail (`belgilar@domen.zona`) → `[EMAIL]`; (2) `+998` va undan keyin 9 ta raqam (raqamlar orasida bitta bo‘sh joy yoki `-` bo‘lishi mumkin) → `[TELEFON]`; (3) aynan 9 xonali alohida son (oldi va ketida raqam yo‘q) — STIR → `[STIR]`. Jami almashtirishlar sonini ham qaytaring.

Imzo: `function mask_pii(string $text): array`

Qaytaradi: `['text' => string, 'count' => int]` — niqoblangan matn va almashtirishlar soni.

MISOL · CHAQIRUV

```
mask_pii('STIR 301234567, tel +998 90 123-45-67')
```

MISOL · NATIJA

```
{"text":"STIR [STIR], tel [TELEFON]","count":2}
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

06

Loyiha himoyasiga tayyorgarlik

Code freeze · v1.0.0 relizi · demo ssenariysi · savol-javob · zaxira reja

v1.0.0 | Demo | Himoya | Zaxira reja

SO‘NGGI 2 HAFTA

Himoyagacha reja

1

Code freeze (−10 kun)

Faqat `fix:` PR lar; yangi imkoniyatlar `v1.1` backlog iga

2

Hujjatlar (−7 kun)

README, ADR lar, Scribe, qo‘llanma, runbook — PR orqali

3

Reliz (−5 kun)

CHANGELOG `[1.0.0]`, `v1.0.0` teg, GitHub Release, production deploy

4

Mashq (−3 kun)

Demo ssenariysi 2–3 marta, vaqt o‘lchanadi, savollar ro‘yxati

5

Zaxira (−1 kun)

Demo video, lokal stek, seed ma’lumotlar, havolalar sahifasi

Himoya kunida kod o‘zgartirilmaydi — faqat ko‘rsatiladi.

RELIZ

v1.0.0: tekshirish, teg va GitHub Release

```
git switch main && git pull --ff-only
composer audit
./vendor/bin/pint --test
./vendor/bin/phpstan analyse
php artisan test

# CHANGELOG: [Unreleased] -> [1.0.0] - 2026-12-21
git commit -am "chore(release): 1.0.0"
git tag -a v1.0.0 -m "OmborPro 1.0.0"
git push origin main --follow-tags

gh release create v1.0.0 --title "OmborPro 1.0.0" \
    --notes-file docs/release-1.0.0.md
```

▶ NATIJA

```
To github.com:jamoa/omborpro.git
 * [new tag]         v1.0.0 -> v1.0.0
https://github.com/jamoa/omborpro/releases/tag/v1.0.0
```

pint --test

faqat tekshiradi, fayllarni o‘zgartirmaydi

tag -a

izohli teg: muallif, sana va xabar saqlanadi

--follow-tags

commit bilan birga izohli teglarni ham yuboradi

gh release

GitHub CLI: Release sahifasi va tavsifi

SAVOL-JAVOB

Komissiya ko‘p beradigan savollar

1 Arxitektura

Nega modulli monolit? Qaysi ADR da yozilgan?

2 Ma’lumotlar

Rezerv va hisob-faktura qaysi tranzaksiyada? Qoldiq manfiy bo‘lmasligi qanday kafolatlanadi?

3 Sifat

Test qamrovi qancha? Qaysi biznes-qoida testlanmagan?

4 Xavfsizlik

Buxgalter buyurtmani o‘chira oladimi? Policy qayerda?

5 Ekspluatatsiya

Xatoni qanday bilasiz? Zaxiradan tiklashni sinab ko‘rganmisiz?

6 Jamoa

Kim nima qildi? Qaysi PR sizniki va review da nima o‘zgardi?

Javob — aniq fayl, test yoki PR ga havola bilan: “`OrderService::confirm`, 42-qator, `OrderFlowTest`”.

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

6-bo‘lim yakuni: Loyiha himoyasiga tayyorgarlik

1

Code freeze dan keyin qanday o‘zgarishlarga ruxsat beriladi?

Savol 1

2

Demo ssenariysi nima asosida quriladi?

Savol 2

3

Demo paytida server ishlamay qolsa, zaxira rejangiz nimalardan iborat?

Savol 3

XULOSA

Asosiy xulosalar

1 Qo‘llab-quvvatlash — eng uzun bosqich

TCO ning katta qismi joriy etilgandan keyin sarflanadi; xizmat to‘rt turga bo‘linadi.

2 Hujjat — kod bilan birga

README, ADR, OpenAPI va qo‘llanma repozitoriyda saqlanadi va PR orqali yangilanadi.

3 SLA va hodisalar jarayoni

P1–P4, runbook va aybsiz postmortem tizim barqarorligini ta’minlaydi.

4 Versiya — bu va’da

SemVer, CHANGELOG va Conventional Commits o‘zgarish ta’sirini oldindan aytadi.

5 Deploy ≠ release

Pennant flaglari imkoniyatni bosqichma-bosqich yoqadi va deploy siz o‘chirishga imkon beradi.

6 Tendensiyalarga ehtiyotkorlik bilan

Avval modulli monolit; low-code va AI — nazorat, ma’lumot xavfsizligi va inson tasdig‘i bilan.

TAKRORLASH

Yakunlashdan oldin sakkizta savol

1. ISO/IEC 14764 dagi to‘rt xizmat turi?
2. ADR ning asosiy qismlari?
3. SLI, SLO va SLA farqi?
4. `2.3.1` da orqaga mos bo‘lmagan o‘zgarish — keyingi versiya?
5. Nega qo‘llangan migratsiya tahrirlanmaydi?
6. Feature flag nimani ajratadi?
7. Qachon modulni mikroservisga ajratish o‘rinli?
8. Mijoz ma’lumotini AI servisiga yuborishdan oldin nima tekshiriladi?

AMALIY MASHG‘ULOT

15-amaliy mashg‘ulot: 5 ta PHP topshiriq va loyiha bosqichi

4 **Reliz cheklistini tekshirish**
Oson · 10 ball

5 **Feature flag: kimga yoqilgan?**
O‘rta · 20 ball

6 **ADR holatlari mashinasi**
O‘rta · 20 ball

7 **CHANGELOG: [Unreleased] bo‘limini tahlil qilish**
Qiyin · 30 ball

8 **SemVer bo‘yicha tartiblash (pre-release bilan)**
Qiyin · 30 ball

Telefon kamerasi bilan skanerlang

PHP kod saytda yoziladi va server tomonidan avtomatik tekshiriladi.

Loyiha havolasini ham shu sahifada topshirasiz; o‘qituvchi baho qo‘yadi.

CHIQISH CHIPTASI

Darsni tark etishdan oldin

Javoblaringizni qog‘ozga yozing va o‘qituvchiga topshiring.

1

OmborPro da oxirgi qilgan o‘zgarishingiz ISO/IEC 14764 bo‘yicha qaysi turga kiradi va nima uchun?

2

`1.2.0` dan keyin bitta `feat:` va ikkita `fix:` commit bo‘lsa, keyingi versiya qanday bo‘ladi?

3

Himoyada jamoangiz uchun eng katta xatar nima va zaxira rejangiz qanday?