Lessons learned — LARP/RPG: Výprava — Rataje nad Sázavou

Sepsáno 2026-08-27, šest dní po odehrání hry (2026-08-21), na konci dávky, ve které se dotahovaly nedokončené sweepy po vyřazení Ctibora a Elišky.

Pro koho: pro tebe, až budeš pokračovat v tomhle scénáři nebo stavět podobnou hru. Není to postmortem hry (ta dopadla dobře) — je to postmortem **způsobu práce s podklady**.

0. Tři čísla, která shrnují celý problém

Metrika	Hodnota
`.md` souborů v adresáři	**432**
z toho `*_backup*.md`	**75**
skutečných dokumentů („rodin" po odstranění `_NN`)	**206**
nejrozvětvenější rodina	`linky_detaily` — **26 verzí**
adresář	`D:\Projects\Git\content\...`
je v gitu	**ne**

432 souborů pro 206 dokumentů. Víc než polovina obsahu na disku jsou historické verze téhož. A leží to v adresáři, který má v cestě slovo `Git`.

To není nadávka na dosavadní postup — bez číslovaných verzí bys neměl historii vůbec. Je to ale přesná diagnóza: **projekt si ručně a draze implementoval verzovací systém, který existuje hotový a zadarmo.**

1. Co fungovalo — tohle si nech

1.1 Tombstones místo mazání

Konvence `~~Ctibor~~ **(VYŘAZENO 2026-08-16)**` je výborná a měl bys u ní zůstat. Smazaný obsah je nedohledatelný a při reaktivaci role se rekonstruuje z hlavy; přeškrtnutý obsah je greppovatelný, vysvětluje *proč* tam něco není, a chrání před tím, aby to někdo „opravil" zpátky. `archiv_vyrazenych_postav_04.md` jako márnice je totéž o patro výš.

1.2 Greppovatelné značky přenosu

`[Ctibor (VYŘAZENO) → Matyáš (fallback)]` · `[hranice M1]` · `[F]` · `(EXTRA)`

Tohle je nejlepší nápad v celém projektu. Značka nese **tři informace naráz** (co se stalo, komu to přešlo, že je to náhrada) a dá se najít jedním grepem napříč vším. `(EXTRA)` u obsahu, který doporučená analýza chtěla škrtnout a kurátor to zamítl, ušetřil minimálně tři kolečka „proč to tu ještě je".

1.3 Sekce „Resilience" u každé linky

Předem sepsané „co se stane, když tahle postava vypadne" bylo prozíravé a **zaplatilo se to dvakrát** (Eliška, Ctibor). Bez toho by vyřazení postavy týden před hrou byla panika, ne rozhodnutí.

1.4 Synchronizační okna jako páteř

Model `W0–W5` (+ přibyla `W2.5`, `W3.5`, `W4.5`) unesl to, že se beaty během léta několikrát stěhovaly. Beat se přesune mezi okny a všechno ostatní drží. Kdyby byl scénář ukotvený na hodinách nebo na kilometrech, každý přesun by trhal celý dokument.

Zároveň to opakovaně chytalo reálnou chybu: **beat naplánovaný pro dvě postavy, které v tu chvíli nejsou na stejném místě.** Těch nálezů („fyzicky nemožné, přesunuto do W3") je v `linky_detaily` nejmíň pět.

1.5 Karty postav psané ve druhé osobě

„Co dnes máš udělat", „Tvé volby", „Meze tvé moci", 🔒 bloky pro pozdní odhalení. Hráč dostane rozkaz, ne popis. Rozdíl mezi *„Klára sleduje Radovana*" a *„Najdi tři konkrétní informace o Radovanovi a rozhodni, které předat*" je rozdíl mezi pasivní a aktivní postavou.

1.6 Rekvizity s ID

`#L1a-C`, `#L5-D`, `#Dite-A` — jednoznačný odkaz na fyzickou věc napříč všemi dokumenty. Kolize `#Deti-*` × `#Dite-*` se našla právě proto, že ID existují a dají se strojově porovnat.

1.7 Psaní povídek jako validační technika

Tohle je zpětně nejužitečnější věc, kterou jsme udělali, a nečekal jsem to. Když se linka přepíše jako souvislá povídka „jak to dopadlo", okamžitě vyplavou strukturální vady, které v beat-tabulce nejdou vidět:

- **L4 má vrchol o okno dřív, než končí.** Ludmilino rozhodnutí padne u brodu (W4) a zbytek dne už jí nic nedělá.
- **L2 má dva protagonisty a dvě různé noci** — bez rozdělení na „Blažej × Vavřinec" a „Vavřinec × Kliment" se nedá napsat jako jedna povídka, protože nemá jeden oblouk.
- **L3 končí antiklimaxem záměrně** a funguje to — ale zjistíš to, až to napíšeš.

Doporučení: **napiš povídky dřív, ne až po hře.** Je to levný test dramaturgie a odhalí věci, které audit tabulek neodhalí nikdy.

2. Co bylo přítěž

2.1 Číslo verze uvnitř odkazu — kořen skoro všech potíží

Když `soubor_A.md` odkazuje na `soubor_B_12.md`, tak **bump B vynutí editaci A**. Editace A vynutí bump A. Bump A vynutí editaci všeho, co ukazuje na A. A tak dál.

Tenhle den to vypadalo takhle:

```
vizualizace_..._03.mermaid  (chtěl jsem jen tohle)
  └─ derivacni_mapa_14 → _15
       └─ derivacni_mapa_agent_07 → _08
            ├─ detske_role_zapojeni_10 → _11
            │    └─ pecete_emitenti_vavrinec_toolbox_05 → _06
            │         └─ rekvizity_listky_* (3×, in-place)
            └─ semanticky_index_hry_10 → _11
                 ├─ karticky_generator_02.py (in-place)
                 ├─ gm_handbook_..._05 ──────────┐
                 └─ tajemstvi_blokujici_linky_04 ─┤
                                                  │
       ...a odsud vede cesta na linky_detaily_16 (master),
       který ukazuje zpátky na tajemstvi_blokujici_linky_04
```

Graf ukazatelů má cykly, takže „bumpni všechno, co ukazuje na bumpnutý soubor" nemá pevný bod. Došlo by to k masterům a od nich zpátky. Každá dávka proto musí někde skončit ručně nakreslenou čarou — a za tou čarou zůstávají zastaralé ukazatele. Dnes jich tam zůstalo devět.

> **Důležité zjištění:** ty cykly **nejsou v datech.** Datová závislost je acyklická (master → index → vizualizace). Cyklus vzniká až v *ukazatelích*, protože `linky_detaily` má navigační poznámku „viz analýza rizik" a ta analýza čte `linky_detaily`. **Je to artefakt čísel v odkazech, ne vlastnost projektu.**

2.2 Táž data ve dvou reprezentacích

`semanticky_index_hry § 2` má **tabulku** beatů a **pod ní prózu**, která říká totéž. Sweep po Ctiborovi 2026-08-16 opravil prózu a tabulku ne — a index si **jedenáct dní protiřečil sám se sebou**, přičemž jeho vlastní hlavička tvrdila, že je opravený.

Stejný vzorec: index duplikuje mastery, derivační mapa duplikuje vztahy, které jsou i v hlavičkách jednotlivých souborů, TL;DR tabulka duplikuje sekci 4 téhož souboru.

Každá duplicita je místo, kde se to rozejde. Není otázka jestli, jen kdy.

2.3 Sloupce s holými ID vs. próza se jmény

Proč sweep selhal konkrétně: sloupec „Postavy" obsahuje `D1, D2, B4, D3, D4`. Próza obsahuje „Ctibor". **Grep na `Ctibor` tabulky minul úplně**, grep na `D4` je utopil mezi desítkami legitimních zmínek („D4 vyřazen", „přenos z D4"). Sweep v dobré víře ohlásil hotovo.

2.4 Anchor mapa s čísly řádků

`derivacni_mapa_agent` obsahuje tabulky „postava → řádek → rozsah". Ty se zneplatní **při první editaci uprostřed souboru** a soubor sám to o sobě přiznává červeným bannerem od 2026-08-16. Užitečný je z té mapy jen sloupec **„Grep pattern"** — ten platí pořád.

Čísla řádků jsou v projektu, kde se soubory týdně mění, čistá režie.

2.5 Skilly s natvrdo zapsanými daty

`skill_mermaid_generator § 3` měl vypsaný seznam 14 názvů subgrafů a pokyn „striktně dodrž". Trasa se změnila, seznam ne. Skill tím **odporoval pravidlu B ve vlastním textu** a při regeneraci musel rozhodovat kurátor, které z jeho dvou pravidel platí.

Derivát zastará hlasitě (regeneruje se, nesoulad je vidět). **Skill zastará tiše** — čte se jednou za pár měsíců.

2.6 Megafiles

`postavy_komplet_19.md` = **2482** řádků, `linky_detaily_16.md` = **2688**. Nedají se přečíst celé kvůli jedné postavě, takže vznikla celá infrastruktura (cost tiery, anchor mapy, grep recepty), jejímž jediným úkolem je obcházet to, že jsou ty soubory moc velké.

> 🔁 **Mimochodem — příklad téhož problému při psaní tohohle dokumentu.** Ta dvě čísla jsem nejdřív opsal z `derivacni_mapa_agent_08.md` (2359 a 2602). Po ověření `wc -l` vyšlo 2482 a 2688. **Agentský soubor, jehož jediným úkolem je vědět, jak jsou soubory velké, měl obě čísla špatně** — a přitom o sobě v banneru poctivě píše, že jeho čísla neplatí. Ručně udržovaná metadata o souborech zastarají vždycky; `wc -l` je zadarmo a nikdy nelže.

2.7 Hlavičky nabobtnalé changelogy

`derivacni_mapa_15.md` má changelog dřív, než začne obsah. Je to cenná historie na špatném místě — čte se při každém otevření souboru, i když ji nikdo nepotřebuje.

3. Doporučení: verzování a kaskády

3.1 Nasaď git. To je celé.

`git init`, `git add -A`, hotovo. Co tím okamžitě zmizí:

Dnešní problém	S gitem
432 souborů pro 206 dokumentů	206 souborů
bump = `cp` + editace + sweep	`git commit`
kaskáda přejmenování	**neexistuje** — jména se nemění
cykly v grafu ukazatelů	**neexistují** — v odkazu není číslo
„která verze to tvrdila?"	`git log -p soubor.md`
„co se změnilo v `_15`?"	`git diff`
changelog v hlavičce	commit message
`*_backup*.md` (75 souborů)	pryč

Odkazy pak zní `linky_detaily.md` a **nikdy nezastarají**. Otázka „musím kvůli tomuhle bumpnout ještě tři soubory?" prostě přestane existovat.

Námitka, kterou čekám: „ale já chci vidět starou verzi vedle nové." → `git show HEAD~3:linky_detaily.md > /tmp/stara.md`. Nebo si nech na jednu konkrétní věc branch.

Skutečné riziko: binárky. `postavy_karticky_07.pdf`, `.webp` mapa, `.svg` rendery — ty do gitu patří jen jako výstupy, ideálně regenerovatelné. 61 MB je pořád v pohodě.

3.2 Když git ne, tak aspoň stabilní jména

Druhá nejlepší varianta, a odpovídá i tomu, na co ses ptal:

```
linky_detaily.md              ← živý, jméno se NIKDY nemění
archiv/linky_detaily_16.md    ← snapshot pořízený při bumpu
archiv/linky_detaily_15.md
```

Odkazy vždycky bez čísla. Snapshot se dělá kopií *do archivu*, ne přejmenováním živého souboru. Tím zmizí sweep i kaskáda a verzování zůstane.

K tvému návrhu psát v odkazech `linky_detaily_xx.md`: funguje to a je to lepší než dnešek — nikdy nezastará. Má to ale dvě mouchy:

- `xx` je nevalidní cesta, takže odkaz nejde otevřít klikem ani ověřit skriptem, že cíl existuje.
- Ztratíš možnost říct „ověřeno proti `_16`", což se občas hodí v changelogu.

Doporučení: stabilní jméno bez čísla (`linky_detaily.md`) je lepší než `_xx` — je to platná cesta, dá se ověřit, a „proti čemu bylo ověřeno" napiš do changelogu tam, kde na tom záleží.

3.3 Rozlišuj závislost od navigace

Cyklus `tajemstvi → linky_detaily → tajemstvi` vznikl smícháním dvou různých věcí:

- **závislost** — „tenhle soubor čte tamten a musí se po jeho změně přegenerovat"
- **navigace** — „mimochodem, existuje k tomu analýza"

Navigační odkaz nemá důvod nést verzi a nemá spouštět nic. Kdybys je značil odlišně (např. `→ zdroj:` vs. `→ viz:`), byl by graf závislostí okamžitě acyklický a šlo by ho topologicky seřadit.

4. Doporučení: duplicity a konflikty

4.1 Deriváty generuj, neudržuj

`semanticky_index § 2` je dnes **ručně udržovaná kopie** beatů z masteru. Proto se rozešel. Kdyby se generoval skriptem z `linky_detaily`, rozejít se nemůže.

Praktický kompromis, když plná generace není reálná: **udržuj jen jednu reprezentaci.** Buď tabulka, nebo próza — ne obojí o tomtéž.

4.2 Napiš si `check.py`

Nejlevnější věc s nejlepším poměrem přínos/cena v celém projektu. Pět assertů, běží za vteřinu:

```python
# 1. počet beat-řádků v indexu § 2 == počet uzlů v diagramu
# 2. žádná (VYŘAZENO) postava není ve sloupci „Postavy" v § 2
# 3. každé #ID rekvizity zmíněné v textu existuje v tabulce předmětů
# 4. každý odkazovaný soubor existuje na disku
# 5. každý mermaid uzel použitý v šipce je definovaný
```

Kontrola 1 a 2 by dohromady odhalily **oba** dnešní nálezy — chybějící `L6.B3b` i jedenáct dní starý nedokončený sweep. Invariant `65 = 65` je už zapsaný v `derivacni_mapa_agent_08.md` jako povinný krok.

4.3 Piš ID i jméno pohromadě

Místo `D1, D2, B4, D3, D4` piš `D1 Vlastimil, D2 Dorota, B4 Matyáš, D3 Přibyslava, D4 Ctibor`. Delší o pár znaků, ale **grep na jméno i na ID najde totéž** a sweep 2026-08-16 by neselhal.

4.4 Skilly popisují postup, ne obsah

Do skillu nikdy nepatří seznam lokací, jmen ani ID. Patří tam pravidlo, jak si je vytáhnout ze zdroje. `§ 3` mermaid generátoru je teď přepsaný přesně takhle.

4.5 Zahoď čísla řádků, nech grep patterny

Anchor mapa s řádky je záporná hodnota — vypadá autoritativně a lže. Grep patterny (`^## A4 –`) platí navždy.

5. Doporučení: design hry

5.1 Agendu nes na funkci, ne na osobě

Vyřazení Ctibora bylo drahé (dva dny práce, dodnes nedotažené stopy), protože jeho agenda byla popsaná jako „co dělá Ctibor". Kdyby byla popsaná jako **role v dramaturgii** („protihlas balady", „deflátor", „nositel jména mrtvého"), přiřadí se jinému hráči přepsáním jednoho řádku.

Projekt na to nakonec přišel sám — přenosová tabulka *funkce → nová postava* je přesně ta správná abstrakce. Škoda, že vznikla až při vyřazování.

5.2 Hlídej zátěž jednoho hráče živě

Matyáš má po kanibalizaci Ctibora **7 teček** v auditu hutnosti — nejvíc ve hře — a `analyza_pov_postav_pocity` ho pořád vede na ★★★☆☆ „s hluchým odpolednem". Audit hutnosti existoval, ale nepřepočítal se s ním.

Doporučení: jedna tabulka „postava → počet beatů → špičky" generovaná ze zdroje, ne psaná ručně. A tvrdý strop (řekněme 5), po jehož překročení se agenda musí někam přelít.

5.3 Cross-group beaty jsou drahé — validuj ko-přítomnost brzo

Nejčastější strukturální chyba v projektu: beat pro dvě postavy, které v tu chvíli nejsou na stejném místě. Objevovalo se to opakovaně, vždycky pozdě, vždycky s poznámkou „fyzicky nemožné, přesunuto".

Stačí jedna tabulka `postava × okno → kde je` a strojová kontrola, že obě postavy beatu jsou ve stejné buňce. Dá se udělat za hodinu a ušetří to opakované přepisování.

5.4 Volby jsou dobré, jen když mají adresáta

Nejsilnější mechaniky ve hře mají **okamžitou viditelnou odezvu**: mísa s opaskem, miskové hlasování, sáček s osudem Lísky. Nejslabší jsou ty, co končí do ztracena — Havel celý den nese

rozhodnutí o táboru bez jediné ozvěny (a projekt to nakonec zaplácl tím, že Havel *zahlédne* předání sáčku, což je dobrá záchrana).

Pravidlo: ke každé volbě dopiš, **kdo a kdy ji uvidí**. Když na to není odpověď, volba není volba, je to poznámka na kartě.

5.5 Tři verze mrtvého je nosná konstrukce

Miskové hlasování MÝTUS / KRYSA / ČLOVĚK s **jasným nositelem každé misky** (Vlastimil / Matyáš / Dorota) je nejlepší finálová mechanika ve hře. Funguje proto, že každá verze má někoho, kdo ji obhajuje ze svého vlastního zájmu — a jeden z nich lže, ale nemůže říct proč.

Zároveň to byla nejkřehčí věc v dokumentaci: tabulka rekvizit ještě 26. 8. psala třetí misku jako „nevím" místo „člověk". *Pasivní zdržení se* místo *třetí aktivní verze* — a Dorotina eulogie by neměla kam dopadnout.

5.6 Postava, o které se jen mluví, je levná a silná

Jíra Hrubý, Janek/Líska, Marika, Kliment, převor — nikdo z nich není na scéně a všichni nesou dramaturgii. **Off-stage postava nestojí hráče, nemá kartu, nemůže improvizovat mimo záměr a nedá se vyřadit.** Když jsi na hraně počtu hráčů, tohle je první směr, kterým škrtat.

6. Checklist pro příští hru

Než napíšeš první scénář:

1. `git init` v den nula.
2. Jeden dokument = jeden soubor, **bez čísla v názvu**.
3. Rozhodni, co je **zdroj** a co **derivát**. Derivát se generuje, nikdy needituje ručně.
4. Postavy značit `ID Jméno` pohromadě, všude.
5. `check.py` s pěti asserty od začátku, ne až když něco praskne.

Během psaní:

6. Skilly = postup, nikdy data.
7. Ke každé volbě dopiš adresáta odezvy.
8. Agendu popisuj jako funkci, ne jako osobu.
9. Strop beatů na postavu, hlídaný generovanou tabulkou.
10. Ko-přítomnost postav validuj strojově.

Před hrou:

11. **Napiš povídky.** Co linka, to jeden oblouk s jedním vrcholem. Co nejde napsat, je špatně navržené.
12. Resilience u každé linky.
13. Tombstones místo mazání.

7. Co udělat teď, když se pokračuje v tomhle scénáři

Seřazeno podle poměru přínos/cena:

#	Krok	Cena	Co to vyřeší
1	`git init` + první commit	5 minut	zastaví růst 432 souborů; od té chvíle žádné další bumpy
2	`check.py` s invarianty 1–2	~1 hodina	odhalí zbylé nedotažené sweepy okamžitě
3	Regenerovat `analyza_pov_postav_pocity` proti masterům	~2 hodiny	**Matyáš je podhodnocený o dvě hvězdičky** — to je věcná chyba v plánování zátěže, ne kosmetika
4	Sjednotit `§ 2` indexu na jednu reprezentaci	~1 hodina	odstraní místo, kde se to naposledy rozešlo
5	Dotáhnout 9 ukazatelů za hranicí kaskády	~1 hodina	jen když se **nedělá** krok 1; s gitem je to bezpředmětné
6	Přepočítat souhrny na 16 hráčů (ne 18)	~1 hodina	`graf_ohodnoceni`, `mapa_postav_vazby` heatmapa, POV průměry

Kdybys měl udělat jen jednu věc, udělej krok 1. Zruší většinu režie, kterou tenhle dokument popisuje, za pět minut práce.

8. Poznámka na závěr

Nic z výše uvedeného neznamená, že dosavadní postup byl špatný. Číslované verze + povinný sweep + tombstones jsou **funkční** systém — udržel projekt konzistentní přes čtyři měsíce, dvě vyřazené postavy, tři restrukturalizace trasy a jeden odehraný herní den. Že se v něm najdou dva nedotažené sweepy, není selhání disciplíny; je to očekávaná chybovost ručního procesu v tom měřítku.

Pointa je jinde: **ten systém řeší problém, který má hotové řešení.** Čas, který šel do `cp`, sweepů a rozhodování, kde ukončit kaskádu, mohl jít do hry.

„Zápis není zločin." — purkrabí Radovan, když se velitel obořil na písaře, že si pořád něco zapisuje.