

# První rodičovský paradox aneb Kdo koho vychovává?

---

Petr Suchý



## Petr Suchý

Jsem původní profesí softwarový odborník, ale také jsem táta a terapeut, který se zabývá výchovou dětí. Nedávno jsem si uvědomil, že výchova dětí a vývoj software má mnoho podobných stránek, a tak jsem nelenil a sepsal pro vás mnoho zajímavých postřehů z této oblasti. Uvědomil jsem si přitom první rodičovský paradox, který dává odpověď na otázku, kdo koho vlastně vychovává? Je to úsměvná myšlenka, která vám poskytne nadhled a odstup při výchově. Přeji vám příjemné čtení.

# Úvod

---

Milí přátelé,

připravil jsem pro vás pár stránek, které ukazují podobnosti mezi výchovou dětí a vývojem software. I když nejste třeba IT odborníci, toto přirovnání vám vnese více světla do toho, co se vlastně při výchově děje.

Začneme například u testování. Jak známo, děti často testují hranice svých rodičů. Znamená to, že zkoušejí, co všechno rodič vydrží. A to je vlastně začátek naší úvahy. Je toto testování součástí výchovy?

Osobně chápu výchovu jako proces, posloupnost výchovných kroků. Na začátku je narozené miminko, na konci by měl být samostatný zdravý dospělý člověk, který má řadu dalších dovedností. Vypadá to tedy, že výchova je proces, který řídí rodič, jenž se snaží vychovat z dítěte pořádného dospělého.

A zde začíná paralela s vývojem software. To je také proces, kdy se utváří aplikace, která má být něčím prospěšná.

Aby se aplikace povedla, musí ji někdo neustále testovat a tím zkoušet, jestli funguje tak, jak má. Cílem je odhalit chyby, které tam programátor nechtěně udělal, opravit je, a tím zajistit, že aplikace je v pořádku.



## Testuje dítě rodiče nebo rodič dítě?



Vypadá to, že klíčové myšlenky jsou podobné. Cílem vývoje software je prospěšná aplikace. Cílem výchovného působení je dospělý, který bude ostatním prospěšný.

Pokud s tím souhlasíte, tak vězte, je to jen část pravdy. Ta důležitější část pravdy je opačná. Děti testují rodiče proto, aby vznikl lepší rodič.

Nedávno jsem na semináři o výchově kluků dostal otázku, jak se pozná hranice, kdy je dobré ještě vyjít dětem vstříc a kdy už ne. Další den při hraní s dětmi jsem si uvědomil, že ony vlastně často jen testují hranice rodičů, tedy zkoušejí, co pro ně rodiče ještě udělají.

Znáte to: “Mami, kup mi lízátko”. Vy řeknete “Nekoupím” a dítě zkusí zesílit tlak, někdy příjemným způsobem - prosí a vyjednává - a někdy nepříjemným způsobem - vzteká se a vydírá. Oba dva způsoby se mohou prolínat a dítě je může stupňovat. Snaží se tak získat, co chce, a někdy se mu to povede. Dostává tedy, co chce, a vytváří novou verzi lepšího rodiče, který je lépe připraven naplnit jeho potřeby. Řadu rodičů to však přivádí do nepříjemných stavů.

Zkusme se podívat na paralelu s vývojem software, která vám může přinést nadhled a úlevu.



# Principy

---

Představte si dítě jako programátora, jenž se snaží vytvořit dobrý softwarový produkt. Softwarový produkt je program, který běží v počítači (hardware). Program se skládá z instrukcí, kterým počítač rozumí, a podle těchto instrukcí provádí operace. V základu se jedná o jednoduché operace sčítání a odčítání. Z nich se dají sestavit složitější instrukce a z nich pak různé programy - hry, účetnictví, webové stránky atd. Prostě všechno to, co vidíte na obrazovce. Program má nějaký účel - například ušetření práce (zpracování dat v počítači má více výhod než na papíře) nebo zábavu.

Když se vrátíme k výchově dětí, tak počítačem (hardwarem) je dospělý člověk (jeho mysl a tělo). A dítě se snaží vytvořit program Rodič.

Maminka se většinou stává rodičem porodem. Od přírody je vybavena základními dovednostmi, takže její program je připravený. Ve hře s panenkami si ho maminka během svého dětství trochu vyzkoušela, ale děti coby reální uživatelé budou určitě vyžadovat určité změny.

Tatínek sice bývá původcem oplodnění, ale opravdovým rodičem se stane až s dítětem. Tatínek má také nějaký základní program, ale ten si v dětství moc nevyzkoušel, a tak většinou není úplně dobře připraven.

Příroda tedy odvádí docela dobrou práci, ale ani ta není bez chyb. Sem tam se stane, že nějaký "rodič" hned po narození dítě odloží do baby boxu. Jsou to naštěstí výjimky, a tak lze říci, že se příroda snaží docela dost. Nicméně dítě čeká ještě spousta práce, než je program Rodič opravdu hotový a v praxi prověřený.

Dítě dostává k dispozici rovnou dva počítače se základními programy. Výhodou je, že když jeden počítač je odpojený nebo se rozbije, zbývá aspoň druhý, který může něco dělat. Nevýhodou je, že mají bohužel jiné hardwarové součástky (hlavně procesor), a tak mohou ve stejné situaci fungovat různě. To se dá sice časem upravit, ale stojí to opravdu hodně a hodně práce. Z pohledu dítěte je lepší využít rozdílů v hardware a připravit trochu jiné programy, které mají různé účely.

A tak se snaží vytvořit program Maminka a program Tatínek. Vy si určitě uvědomujete rozdíly v hardware (ty jsou koneckonců vidět na první pohled), ale možná ne v těch programech, které v nás fungují. A tyto rozdíly často způsobují konflikty mezi partnery. Například program Tatínek je v určitých situacích benevolentní (nechá dítě vlézt do studeného potoka), zatímco program Maminka je více pečující (někdy dítě může rozmazlovat). Pokud byste se chtěli dozvědět více o rozdílech v těchto programech, podívejte se na záznamy webinářů Výchova kluků a Výchova dívek.

# Verzování

---

První dítě vytvoří první verzi rodiče. Druhé dítě, které přišlo na svět později, dostává lepší verzi, ta je totiž lépe otestovaná a připravená na více situací. A tak druhé dítě pokračuje v započaté práci.

Občas se stává, že programátor obsadí veškerou kapacitu dostupného hardware, a tak byt' je první verzi odladěná, kapacita je zaplněná a další programátor zde už nemá co dělat. Druhé dítě tak vůbec nepřijde na svět. Výhodou je, že programátor má hardware jen pro sebe, nevýhodou je více práce.

Běžně ovšem druhé dítě pokračuje v testování a první dítě mu pomáhá. Výsledný produkt (software) se zlepšuje - je schopen řešit lépe situace s druhým dítětem, ale už také umí řešit úlohy párové, kdy ho testují oba zároveň. Vytvoří tak společně druhou verzi produktu.

Někdy se objeví třetí dítě, které už dostává velmi dobrý produkt a není téměř co vylepšovat. Teprve v době, kdy třetí dítě hodně interaguje s prvními dvěma dětmi, společně zjišťují, že v produktu se objevují další defekty vyplývající z nových situací. Společným úsilím tak směřují k další verzi. To se může opakovat teoreticky dále, ale většina HW mívá v tuto dobu už úplně zaplněnou kapacitu a produkt už nelze moc zlepšovat z hlediska plnění paralelních úloh. Většina programátorů se pak věnuje vývoji a ladění specifických funkcí.

# Sady instrukcí

---

Představte si dítě jako programátora, který se snaží vytvořit dobrý softwarový produkt, což je program, který běží v počítači. Program se skládá z instrukcí, kterým počítač rozumí, a podle nich provádí operace. Program má nějaký účel - například ušetření práce nebo zábavu.

Jsou dvě základní sady programových instrukcí. První sada je DEJ MI TO a druhá sada je DĚLEJ NĚCO. Ukazuje se, že s těmito dvěma sadami dokážou mladí programátoři zázraky.

Sada DEJ MI TO obsahuje další specifické instrukce - PODEJ MI TO, PŮJČ MI TO, KUP MI TO atd. Tyto instrukce samy o sobě by nic moc nezpůsobily, je potřeba k nim přidat objekt zájmu. Například ČOKOLÁDU, ŽVÝKAČKY, AUTO, PANENKU,.....Celá instrukce pak zní DEJ MI ČOKOLÁDU, KUP MI AUTO atd.

Objekty lze rozdělit do tříd a ty do podtříd. Třídy jsou například JÍDLO, PITÍ, HRAČKY, TECHNIKA atd. Podtřídy si sami jistě dokážete doplnit, stačí uvést, že velmi oblíbená podtřída jsou SLADKOSTI, MOBIL a POČÍTAČ.

Podstatné je, že občas není možné splnit instrukci perfektně, ale stačí ji splnit přibližně a to tak, že nahradíme objekt dané podtřídy jiným objektem stejné podtřídy, např. někdy stačí dodat bonbóny místo čokolády, a programátor je spokojen. Ovšem mezi třídami už to nefunguje, například dodat místo čokolády bramborovou kaši, už není považováno za splnění pokynu.

Ukazuje se, že některé děti jsou méně náročné na kvalitu software a stačí jim splněná instrukce s dostatečně vhodným objektem. Některé však vykazují sklony k perfekcionismu a snaží se software donutit k přesnému vykonávání instrukcí. Kombinace obou přístupů je fajn, protože vytvořený software je stabilní a vhodný i pro další uživatele.

Bohužel se vyskytují i programátoři s proměnlivým přístupem ke kvalitě. Jednou se spokojí s málem, někdy jsou urputní, a dožadují se perfektního výsledku. Když to dělají dobře, dokáží vyladit software tak, že ten pozná i tyto výkyvy, a dojde k vysokému souladu programátora a jeho software. Nevýhodou je, že toto individuální naladění snižuje přenositelnost, a tedy při použití jinými uživateli vyvolává u těchto zmatek a nespokojenost.

Druhá sada instrukcí DĚLEJ NĚCO obsahuje další specifické instrukce - HREJ SI, PUSTĚ MI, POJĚ VEN, VEZMI NA VÝLET. Tyto instrukce opět manipulují s třídami a podtřídami objektů např. PUSTĚ MI FILM (podtřída Harry Potter).

Zvláštní je, že někdy může být objektem sám programátor např. HREJ SI SE MNOU. Tato kombinace instrukce s objektem bývá energeticky velmi náročná a málokterý produkt dokáže vykonávat tuto funkci nepřetržitě delší dobu. Důvodem obvykle je, že tento objekt (sám programátor) je značně nestabilní a při běhu této funkce zadává další a další detailní instrukce, čímž obvykle zahltní už tak vyčerpanou kapacitu HW. Časem se však programátor poučí a tuto funkci přestane vyžadovat. Místo toho nastupuje nová instrukce, která je odvozena z instrukce DEJ MI, má však opačný charakter. Dá se chápat jako DEJ MI ČAS A PROSTOR, většina ji zná pod názvem NECH MĚ BÝT. Objevují se variace jako NECH MNE SI HRÁT, NECH MNE JÍT VEN, DEJ MI POKOJ, NESTAREJ SE atd.

Tento přechod programátora na vyšší úroveň je vítaná změna. Hardware má čas regenerovat, případně více pracovat na jiných úlohách, které zajistí do domácnosti více zdrojů. Jak programátoři rostou, tak požadují méně objektů, ovšem výrazně dražších. Kde dříve stačilo Lego za 300 Kč, je nyní mobil za 3000 Kč. Dá se tedy říci, že programátor i program se časem vyvíjejí společně. Zajímavé je, že zde funguje zpětná kompatibilita a takový program je schopen obsloužit starší i mladší uživatele.

# Životní cyklus

---

Ukázali jsme si, že sice zdánlivě rodiče vychovávají děti, ale také děti vychovávají rodiče. Učí je poslouchat určité příkazy (DEJ MI TO, UDĚLEJ TO) a pak je vesele a donekonečna testují.

Je opravdu báječné pozorovat malé programátory při práci. Když je vidíte, co všechno dokáží těmito jednoduchými příkazy zařídit (například získat bonbony před obědem nebo se nechat kompletně obléci), tak máte pocit, že v tom musí být nějaké kouzlo. Zázrakem přírody je, že ze začátku to dokonce dokáží beze slov. Vyvíjený produkt nejčastěji kategorie MATKA dokáže poskytovat funkce, i když instrukce nejsou kódovány jazykem, ale spíše těžko rozpoznatelnými zvuky, někdy i mimosmyslovými signály. Hodně malí programátoři totiž ještě neumí jazyk používat a přesto to funguje. Stačí zavrňt, zakníkat a matka už běží. A někdy stačí, když dítě jen pomyslí na určitou věc, a citlivější matky to vnímají a jdou se podívat, jestli je vše v pořádku. Jakmile se dítě jazyk naučí používat, je jeho programování mnohem zajímavější, kreativnější a pestřejší.

Čím je programátor starší, tím je produkt vyspělejší a má více funkcí. Paradoxně je ale po několika letech přestává poskytovat. Tento paradox z hlediska programátorů nebyl dosud uspokojivě vysvětlen. Je možné, že dochází k vyčerpání kapacity hardware, který sice běží dále, ale většinu kapacity spotřebuje na údržbu sama sebe. V tom případě programátor využívá svých schopností a začíná se rozhlížet po jiném HW, mladším a výkonnějším. Tak často nalézá nový HW pod názvem PARTNER(KA). Nový HW vypadá mnohem lákavěji než původní, bohužel však časem programátor zjistí, že je obsazen spoustu jiných programů, které se velmi těžko upravují. Občas se tedy vrací k původnímu HW, který sice není tak výkonný, je ale spolehlivý, a je jasné, co se od něj dá očekávat, byť toho není mnoho.

# Ladění výkonu

---

Kdyby byl rodičovský hardware opravdu kvalitní, tak by to měli programátoři docela lehké. Bohužel však hardware má řadu biologických komponent a není dokonalý. Jeho výkonnost je proměnlivá. Tak se ukazuje při testování produktu, že některé instrukce při opakování začínají rychle selhávat. Například rychlé opakování instrukce DEJ MI ČOKOLÁDU obvykle přestává fungovat, proto programátoři používají tzv. rozšíření základních sad instrukcí.





## **Využívání emocí při ovládnání rodičů mívá úspěch.**

Běžným rozšířením jsou EMOCE. Velmi účinný je například VZTEK, který se dá dokonce škálovat od NAŠTVANOST přes KŘIK až po VÁLENI PO ZEMI. Takto aktivované rozšíření často velmi posiluje význam základní instrukce. Takže instrukce DEJ MI ČOKOLÁDU doprovázená výbuchem vzteku má větší naději na úspěch, zejména v obchodě. Rodič se bojí, že bude označen okolím za nekvalitního rodičovský produkt, a tak rychle kupuje čokoládu a opouští obchod. Bohužel některé produkty nebyly dobře otestovány a při použití těchto rozšíření vykazují značné defekty (například se brání křikem nebo odmítají zatvrzele instrukce splnit), což může vyústit ve zklamání programátora, a ten časem od těchto rozšíření upouští.

# Užití cyklů

---

V programátorských jazycích se často používají cykly. V našem případě je jasné, že nekonečný cyklus nemá praktické použití, protože vyčerpá velmi rychle kapacitu HW. Obvykle používaný cyklus je BUDU TO OPAKOVAT, DOKUD NEDOSTANU, CO CHCI. Někteří programátoři jsou s tímto cyklem velmi úspěšní. Je ale řada rodičů, která je na tento cyklus alergická, a vyvolává to v nich velké podráždění. Toto podráždění spotřebovává velmi rychle dostupnou kapacitu hardware, a tak se ukazuje, že není moc účinný. Programátoři by rádi preferovali výkonnější hardware, bohužel však na jeho výběr nemají žádný vliv.

# Operační systémy

---

V minulosti docházelo k velmi vysoké chybovosti při vývoji. Ty se projevovaly tak, že se HW místy vzbouřil a mohlo to skončit dokonce tak, že programátor byl fyzicky potrestán. Odborně se tomu říká zpětná vazba. Znamená to, že HW nesnesl další zátěž a projevil obvykle nejdříve slovní a někdy i fyzický akt násilí, který vedl k tomu, že programátor snížil snahu HW ovládnout a přizpůsobit si ho.

Docházelo i k paradoxním situacím, kdy HW se vzbouřil úplně, snažil se ovládnout programátora a vychovat z něj poctivého a pracovitého člověka. Výzkumy však prokázaly, že přílišné násilí ve zpětné vazbě nevede dobrým směrem. Místo poctivého a pracovitého člověka vyrůstaly tyto děti buď ve zlomené osobnosti, nebo utíkaly z domova nebo převzaly násilnický program rodiče a předávaly ho do další generace. Ukázalo se, že vztah mezi programátorem a HW musí být podroben hlubokému zkoumání, aby bylo vyjasněno, co všechno se v tomto vztahu děje. Byla objevena spousta dobrých myšlenek a nápadů, jak tento vztah zlepšit a dosáhnout většího souladu mezi programátorem a HW. Tyto principy byly zformulovány do různých konceptů, které programátorům pomáhají s jejich prací.

Tyto koncepty se instalují na HW jako operační systém, nad kterým se programuje mnohem lépe. Je zajímavé, že stejně jako máme různé počítačové operační systémy například Windows, Linux, iOS, Android atd. tak máme různé rodičovské operační systémy jako RESPEKTOVAT a BÝT RESPEKTOVÁN, MONTESSORI PRINCIPY nebo VÝCHOVA BEZ PORAŽENÝCH. Pokud byste se ptali, který je nejlepší, tak na to není jednoduchá odpověď. Podobně jako Windows má své příznivce, Linux své nadšence a další své vyznavače, tak i tyto systémy mají různé zastánce. Ukazuje se, že každý systém má své výhody a nevýhody. Někteří rodiče doufají, že instalací všech těchto operačních systémů na svůj HW dosáhnou skloubení výhod a minimalizaci nevýhod, ale zatím to není prokázáno. Naopak může dojít k zahlcení HW přílišnou zátěží a pak nejsou programátorům k dispozici vůbec (perfekcionistický rodič může sám sebe dostat do vyhoření tím, jak se snaží naplnit všechny možné představy o správném rodičovství).

Tak jako u počítačů bude asi platit, že rozhodující je kompatibilita HW s operačním systémem (Linux na iPhone se instaluje blbě). Takže je dobré nechat na rodiči, ať si sám dojde k tomu, který operační systém je pro něj nejvhodnější, protože je to z pohledu programátora nejjednodušší. Instalace sice trvá poměrně dlouhou dobu, ale nakonec z toho těží jak programátor (HW se ovládá mnohem lépe), tak i rodič (jeho zpětná vazba směrem k programátorovi je adekvátní a prospěšná). Je tak dosaženo většího souladu a rozvoj obou jedinců může probíhat optimálně.

## Závěr

---

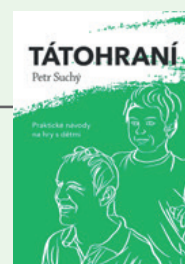
Doufám, že vám tento náhled na snažení dětí přinesl novou perspektivu jak přistupovat k výchově. Jejich testování hranic berte jako pokusy o vytvoření dobrého produktu z jejich pohledu. Zasmějte se tomu společně a dejte jim mírnou zpětnou vazbu. Oni stejně tuší, že vy už jste dávno naprogramováni a jejich testování na tom nic moc nezmění. Jedině snad v místech, kde budete chtít vy sami.

### Chcete více inspirace?

---

Webináře Výchova kluků přináší jednoduchý koncept na výchovu. Vrací nás k přirozenému chápání klukovských potřeb a možností jejich naplňování. Tak je možné snadno podporovat rozvoj kluků a jejich přerod ve

zdravé a plnohodnotné muže. Rodičům se po jejich shlédnutí často uleví. Dostanou dobré triky, jak řešit s kluky konflikty, domácí povinnosti, jak přistupovat k introvertům a citlivějším klukům, a jak si rodičovství zjednodušit.



[Více zde](#)

### Co je to Tátohraní?

---

V rámci Tátohraní podporuji vztahy tátů a dětí. Pořádám akce pro táty a děti – Tátovýlety, Apalucha, Dobývání hradu, vodácké výlety...Připravuji nastávající táty na jejich roli. Tátům v problémech (práce, rodina, zdraví, vztahy) radím, jak z toho ven. Soustředím se na výchovu kluků a radím maminkám, jak na to. Mým cílem jsou spokojení a fungující tátové, pečující a láskyplné maminky, zdravé a veselé děti.

[Více zde](#)