

Porovnání SAFe s Agilním manifestem a alternativy škálování agility

SAFe vs. hodnoty a principy Agilního manifestu

Agilní manifest (2001) formuloval 4 hodnoty a 12 principů, které kladou důraz na **jednoduché, adaptivní procesy**, posílení týmů a rychlé dodávání hodnoty zákazníkovi. Naproti tomu **Scaled Agile Framework (SAFe)** je **rozsáhlý rámec** (od 2011) navržený k aplikaci agilních postupů v celopodnikovém měřítku. SAFe čerpá z agilního vývoje, Lean managementu a systémového myšlení a poskytuje **strukturované postupy, role a artefakty** pro koordinaci mnoha týmů v rámci velké organizace.

SAFe se **hlásí k agilním hodnotám**, ale jeho způsob realizace se od původního agilního manifestu v lecčems liší:

- **Procesy vs. lidé:** Agilní manifest preferuje „*jednotlivce a interakce před procesy a nástroji*“. SAFe sice zdůrazňuje potřebu *Lean-Agile leadershipu* a spolupráce, avšak zavádí **mnoho nových rolí, úrovní a ceremonií**. Kritici poukazují, že tím vzniká **předepisový „byrokratický“ aparát**, který může agilní kulturu potlačovat. Např. společnost ThoughtWorks od roku 2015 **nedoporučuje SAFe**, protože nadměrná standardizace v praxi brání skutečné agilitě (týmy ztrácejí autonomii a tvořivost). Autor Lean UX Jeff Gothelf po zkušenostech s klienty shrnuje: „*SAFe není agilní*“ – vnímá, že rámec předepisuje rigidní pravidla namísto podpory agilních principů.
- **Dokumentace vs. fungující produkt:** Manifest klade „*fungující software před rozsáhlou dokumentací*“. SAFe zdůrazňuje **vestavěnou kvalitu** a potenciálně dodávatelný increment v každé iteraci i na programové úrovni. K dosažení sladění však SAFe zavádí artefakty jako strategické roadmapy, program boardy a řadu plánovacích workshopů. To může vést k **silnějšímu důrazu na centralizované plánování** (Program Increment plánování na 8–12 týdnů) oproti pružnému dodávání malých funkčních celků. Jak trefně poznamenal jeden praktik: při doslovné implementaci „*SAFe zajišťuje vodopád mini-iterací podle příručky, ale s malou skutečnou hodnotou*“ – naráží tak na riziko, že týmy v SAFe mohou jet podle plánu spíše formálně než opravdu iterativně přinášet hodnotu.
- **Spolupráce se zákazníkem:** Manifest preferuje „*spolupráci se zákazníkem před vyjednáváním smluv*“, tedy přímou a častou zpětnou vazbu. SAFe zapojuje zákaznickou perspektivu nepřímou – přes **produktový management, Product Ownery a občasné Product Steering komise**. Zpětná vazba se typicky sbírá v rámci Sprint review a na úrovni Program Incrementu (např. *System Demo* po každém PI). V **mnohohvrstvé struktuře** SAFe ale hrozí, že týmy budou od skutečných koncových uživatelů dál. Naproti tomu například LeSS nabádá, aby **týmy samy komunikovaly se zákazníky** a nespolečaly jen na zprostředkování Product Ownerem. SAFe se snaží zákaznický hlas zohlednit přes role a backlogy, ale není tak *bezprostřední*, jako když malý Scrum tým spolupracuje přímo s objednatelem.
- **Reakce na změnu:** Manifest upřednostňuje „*reagování na změny před dodržováním plánu*“. SAFe pracuje s **časově fixními programovými intervaly (PI)**, během nichž se

týmy snaží splnit cíle PI plánu. Změny priorit se tak v SAFe řeší spíše *iterativně* – např. každé čtvrtletí při plánování dalšího PI, než *průběžně v každém sprintu*. SAFe ovšem obsahuje i mechanismy pro agilní adaptaci: mezi PI cykly probíhají *Inspect & Adapt* workshopy k reflektování a úpravám plánů. Navíc na týmové úrovni SAFe operuje klasickými sprinty. *Formálně* tak umožňuje reagovat na změny, ale kritici tvrdí, že fixní kadence a hierarchie mohou **zpomalit okamžitou reakci**. Steve Denning dokonce uvádí, že SAFe „dává managementu mandát nazývat se *agilními*, zatímco dál dělají, co vždy“, a ptá se „*proč by někdo s opravdovým agilním mindsetem vůbec používal SAFe?*“, přičemž SAFe označuje za „*ztělesnění fake agile*“.

Shrnutí: SAFe se od agilního manifestu odchyluje **mírou formálnosti** – zatímco Agilní manifest propaguje co nejlehčí rámec a evoluční vznik procesů, SAFe předem definuje relativně **komplexní recept** na škálování agility. Podstatou sporu je, zda je onen recept nutný k udržení agility ve velkých firmách, nebo zda právě odporuje agilní filozofii. SAFe klade důraz na **sladění strategie a vývoje (alignment)** napříč velkou organizací a přidává „*just enough*“ procesů a struktur, aby stovky lidí pracovaly koordinovaně. Oproti tomu agilní manifest klade důraz na **empirii, sebeřízení malých týmů a organické přizpůsobení** – tedy spíše *minimalistický přístup*.

Klíčové rozdíly v přístupu k agilitě, škálování a implementaci

Přístup k agilitě: Tradiční agilní metody (Scrum, XP aj.) předpokládají *malé týmy* (cca 7±2 členů) a dávají jim značnou autonomii. Škálování agility na desítky či stovky týmů proto není v Agilním manifestu explicitně řešené. SAFe tuto mezeru vyplňuje tím, že navrhuje **víceúrovňový model**: týmy jsou seskupeny do **Agile Release Trainů (ART)** na programové úrovni a dále do portfolií na nejvyšší úrovni. Každý ART sdružuje ~5–12 týmů, které synchronizují své sprinty a společně dodávají *integrovaný increment* funkcionality v rámci Program Inkrementu (obvykle 5 sprintů + plánovací a inovační interval). **Koordinace** je zajištěna přes eventy jako PI Planning (celý vlak plánuje společně) a přes role jako **Release Train Engineer (RTE)**, který plní roli „Scrum Mastera“ pro celý vlak. Tento *top-down* koordinovaný cyklus **zajišťuje sladění**, avšak může omezit volnost týmů měnit směr uprostřed běhu – týmy se zavazují k cílům PI a větší změny řeší až v dalším PI.

Přístup k škálování: SAFe předpokládá, že organizace *potřebuje* určitou strukturu pro škálování agility – přidává proto **vrstvy procesu a role**, aby bylo jasno *kdo za co zodpovídá* v mnohotýmovém prostředí. Naproti tomu alternativní přístupy jako LeSS či Nexus volí *opačnou strategii*: **co nejméně přidávat navíc** a raději *zjednodušit organizaci* (např. zrušit složité oddělení podle specializací) než přidávat nový proces. LeSS autoři přímo říkají: „*nesnažte se škálovat Scrum, raději deskalujte organizaci*“. Jinými slovy, kde SAFe buduje rámec okolo stávající velké firmy, LeSS či Nexus se snaží **přetvořit firmu tak, aby fungovala jako jedna velká verze Scrum týmu**. SAFe tak reprezentuje *inženýrský přístup* ke škálování (definujeme architekturu procesu), zatímco agilní puristé prosazují *evoluční přístup* (nechtě si týmy najdou cestu skrze inspekci a adaptaci v menších celcích).

Implementace a zavádění: Způsob nasazení SAFe v organizaci bývá **plánovitý a řízený shora**. Typicky probíhá školení managementu i týmů, ustanovení nových rolí (RTE, Product Manager, System Architect aj.) a následné „**rozjetí**“ **Agile Release Trainů** najednou (tzv. *SAFe*

Quickstart, kde se několik týmů naráz zapojí do prvního PI Planningu). Tato *big-bang* transformace může být riskantní, ale má za cíl rychle nastolit jednotný postup. Úspěšné transformace uvádějí, že je klíčové mít podporu vrcholového vedení a interní LACE (Lean-Agile Center of Excellence) pro koučování průběhu. V praxi se organizace někdy rozhodnou zavádět jen **části SAFe** (např. jen úroveň Program = tzv. *Essential SAFe*). To umožňuje postupnou adopci – např. začít koordinovat několik týmů společným PI a teprve časem přidat portfolio řízení.

Naopak **agilní manifest** a jeho principy preferují *postupné, emergentní* zlepšování. Mnoho agilních koučů proto varuje před *příliš rigidní implementací* jakéhokoliv frameworku. Ken Schwaber (spoluautor Scrum) kritizoval firmy, které „si myslí, že **rámec odvede transformační práci za ně**“ – podle něj skutečná agilita vyžaduje změnu myšlení, ne jen nasazení procesu. Rizikem SAFe tedy je, že firma **“pověsí plakát SAFe”**, ale nezmění firemní kulturu (pak jde jen o *názvy bez skutečné agility* – viz kritika *fake agile* výše). Úspěšná implementace SAFe vyžaduje, aby lidé pochopili **agilní principy** pod ním: i tvůrce SAFe Dean Leffingwell říká, že „*nemůžete škálovat bordel*“, čímž zdůrazňuje nutnost dobrých týmových praktik – bez nich SAFe nepomůže. Někteří experti (např. James Grenning) uznávají, že SAFe by *mohl* posloužit jako **odrazový můstek** pro tradiční firmy: „*Pokud SAFe je to, kde se naučíš dělat agilně, abys pak zjistil, jak být agilní, možná to pomůže*“. Tato pragmatická perspektiva vidí SAFe jako *tréninkové kolečko*, které organizaci nasměruje k agilitě – ovšem s rizikem, že firma ustrne u koleček a nikdy se jich nezbaví.

Názory a argumenty pro a proti SAFe

V agilní komunitě vyvolává SAFe **živé diskuse**. Níže jsou shrnuty hlavní argumenty zastánců i odpůrců SAFe, tak jak je prezentují respektované autority:

Argumenty PRO SAFe (obhajoba)

Škálovatelnost a sladění: SAFe poskytuje ucelený rámec pro koordinaci mnoha týmů na strategických cílech. Pomáhá velkým organizacím **sjednotit vývoj s byznysem** a odstranit chaos při řízení stovek lidí. Díky **pravidelným kadencím a synchronizaci** (PI Planning, Scrum of Scrums aj.) všichni chápou aktuální směr a mohou na něm spolupracovat.

Ověřené výsledky ve velkých firmách: Řada společností reportuje díky SAFe zlepšení. Například **FedEx** po zavedení SAFe zaznamenal **výrazný růst engagementu a spokojenosti zaměstnanců**.

Handelsbanken (Švédsko) díky SAFe zkrátil průměrný čas vývoje nových funkcí o **30 %** a významně zlepšil schopnost prioritizace napříč portfoliem. I další firmy (Mercedes-

Argumenty PROTI SAFe (kritika)

Byrokracie a hierarchie: Kritici tvrdí, že SAFe zavádí těžkopádnou strukturu podobnou starému waterfall modelu. Ken Schwaber s nadsázkou psal: „*Hoši z RUP jsou zpět... dusí agilitu SAFe*“. ThoughtWorks pozoruje, že SAFe **vytváří organizační tření, podporuje síla, generuje plýtvání a omezuje autonomii a experimentování** – tedy přesný opak toho, co agile zamýšlí.

„Fake Agile“ – rámec místo mindsetu: Mnozí agilní experti označují SAFe za *pouhou iluzi agility*. Steve Denning píše, že SAFe „*ničí samotnou podstatu Agilu*“ a umožňuje managementu pokračovat v mikromanagementu pod hlavičkou agility. Podobně Martin Fowler a ThoughtWorks v Tech Radaru varují, že SAFe adopce často vedou k **přehnané standardizaci namísto**

Argumenty PRO SAFe (obhajoba)

Benz, Porsche, Nordea...) uvádějí úspěšné transformace a vyšší **rychlost dodávání hodnoty** se SAFe.

Komplexní end-to-end rámec: Zastánci vyzdvihují, že SAFe pokrývá **celý hodnotový tok** – od strategie a investic (portfolio level) přes koordinaci programů až po týmovou realizaci. To pomáhá překlenout propast mezi vedením a týmy. Např. konzultanti z IBM poznamenali, že jejich velcí klienti často selhali s čistým Scrumem, protože **neřešili závislosti mimo tým**; tito klienti by „*profitovali, kdyby hned na začátku využili prvky osvědčeného rámce jako Deanova SAFe*“. SAFe také integruje Lean praktiky (např. Kanban, systémovou optimalizaci toku) a klade důraz na technickou excelenci (*Built-in Quality*), čímž se snaží zachovat inženýrskou kvalitu při škálování.

Adaptabilita a evoluce: Tvůrci SAFe tvrdí, že rámec není dogma – lze ho *přizpůsobit*. SAFe má 4 konfigurace (Essential, Portfolio atd.) a doporučuje začít v malém. Důraz klade na *Inspect & Adapt* cykly a kontinuální učení na všech úrovních. Příručky SAFe zdůrazňují, že kultura *neustálého zlepšování* je klíčová a že týmy mají experimentovat (např. zavádět **Komety inovací** – Innovation sprints – pro zkoušení novinek). Při správném mindsetu tak SAFe firmy *evolučně zlepšují* sprint od sprintu, PI od PI.

Argumenty PROTI SAFe (kritika)

skutečné agility. Jedovatě to shrnuje Jim Highsmith (spoluautor Manifestu): „*Problém SAFe je, že je safe (bezpečný). Bezpečí neplodí inovace.*“ (citováno v).

Přílišná komplexnost a rigidita: Odpůrci namítají, že *přidávat další vrstvy procesů není řešení, že „méně je více“*. LeSS přímo radí „*neskalujte, pokud nemusíte*“. SAFe podle kritiků formalizuje to, co by mělo organicky vznikat retrospektivami. Přidává tolik pojmů (ART, RTE, Epics, Solution Train...), až hrozí, že týmy tráví více času procesem než skutečnou kreativitou. Jeden agilní kouč popisuje zkušenost: „*V SAFe jsme jen vyplňovali šablony a účastnili se ceremonií, skutečná hodnota kulhala.*“ Také **náklady na zavedení** (školení stovek lidí, konzultanti) jsou vysoké – Ken Schwaber varuje manažery, aby si ověřili, zda investice do SAFe přináší reálné zlepšení.

Neúspěšné adopce a kritika autorit: V agilní komunitě je pozoruhodné, že **mnoho respektovaných osobností veřejně SAFe odmítá**. Patří mezi ně i spolutvůrci agilního manifestu: Ron Jeffries, Andy Hunt či Brian Marick – všichni prohlásili, že „*SAFe není agilní přístup*“. Andy Hunt dokonce s nadsázkou poznamenal: „*Mí přátelé si udělali skvělou kariéru uklízením škod po katastrofálních zavedeních SAFe.*“. Také velcí hráči jako U.S. Air Force **odmítli SAFe** – její šéf softwaru Nicolas Chaillan v 2019 označil SAFe za „*Trojan horse*“, který dává falešný pocit bezpečí, a zakázal ho ve svých týmech s tím, že „*SAFe je anti-DevSecOps*“. I když je SAFe populární (nejrozšířenější framework dle State of Agile 2020), tato popularita pramení spíše z přízné manažerů než z podpory agilních odborníků.

Poznámka: Výše uvedené pro a proti ukazují, že SAFe **rozděluje komunitu**. Zastánci uznávají, že není dokonalý, ale vidí v něm **praktický způsob, jak zavést agilitu v komplexním podnikovém prostředí**. Odpůrci namítají, že tím se vytrácí duch agility a **flexibilita**, již malé týmy mají. Pravděpodobně nejzdravější přístup je vnímat SAFe jako *jeden z možných nástrojů*, který může pomoci v určitých kontextech – avšak úspěch závisí na **agilním myšlení lidí**. Bez něj může i sebelepší framework selhat.

Případové studie: úspěchy a neúspěchy SAFe

Úspěšné implementace: Mezi často citované příklady úspěchu SAFe patří:

- **FedEx** – Globální logistický gigant zavedl SAFe již kolem roku 2013. Do roku 2020 vytvořili přes 300 agilních release trainů napříč společnostmi. FedEx uvádí, že transformace přinesla **vyšší zapojení a nadšení zaměstnanců**, kteří si nový způsob práce oblíbili, a to vedlo i ke zlepšení výkonnosti. Podle CIO FedExu „*když lidé, kteří odvádějí práci, to milují a jsou zapálení, vše ostatní se postará samo*“. Zároveň FedEx v období transformace výrazně rostl (během pandemie +25 miliard USD za 2 roky), což firma částečně připisuje zvýšení agility.
- **Handelsbanken** – Tradiční banka (založena 1871) se díky SAFe dokázala udržet na špičce inovací ve finančním sektoru. Po zavedení SAFe uvádí zkrácení času dodání nových funkcí o **30 %** a zlepšení kontinuity dodávek a souladu s portfoliem o více než **50 %**. Banka rozšířila agilní způsob práce na 24 ART a proškolila tisíce zaměstnanců, přičemž si udržela svou *decentralizovanou kulturu* – SAFe se jí podařilo sladit s firemní filozofií lokální autonomie poboček.
- **Mercedes-Benz** (případová studie SAI) – Tradiční výrobce automobilů použil SAFe k **digitální transformaci**: posunul důraz od mechanického inženýrství k softwaru. SAFe pomohl rozbít silové uspořádání a nastartovat *kulturní změnu*, nutnou pro agilitu v IT oddělení.
- **Others**: Další publikované úspěchy zahrnují např. **Nordea** (200letá banka, díky SAFe lídr v digitálních inovacích), **Porsche** (sjednocení vývoje vozového softwaru a IT) či **British Telecom**. Tyto studie (často publikované přímo Scaled Agile, Inc.) vykazují společné přínosy: **zrychlení dodávek, lepší sladění byznys-cílů s IT, zvýšení transparentnosti a vtažení zaměstnanců do dění**.

Je fér dodat, že *pozitivní případové studie často pocházejí od samotných propagátorů SAFe*, takže zdůrazňují úspěchy a měřitelné benefity. Měřit dopad transformace není snadné – firmy obvykle sledují ukazatele jako **time-to-market**, **NPS zaměstnanců**, kvalitu či obchodní výsledky, a pokud se tyto zlepšují, dávají to do souvislosti s agilní iniciativou.

Neúspěšné implementace: O nezdarech se veřejně mluví méně (firmy se s neúspěchem nerady chlubí), ale z doslechu a komentářů expertů plyne, že **mnoho zavádění SAFe selhalo či nepřineslo očekávané výsledky**.

- Spolutvůrce Agile Manifestu **Andy Hunt** upozornil, že zná lidi, kteří si “udělali kariéru” tím, že **přicházeli napravovat neúspěšné adopce SAFe** u klientů. To naznačuje, že *nezanedbatelný počet* velkých transformací s SAFe skončil **deziluzí a nutností návratu k základům**. Časté příčiny nezdaru zahrnují: setrvávající stará kultura (pouze se obalí SAFe terminologií), nedostatečné zaškolení nebo podpora shora, či snaha “dělat SAFe” bez pochopení *proč*.
- Z veřejných vyjádření víme, že např. **Nicolas M. Chaillan** (býv. Chief Software Officer, U.S. Air Force) po zkušenostech označil SAFe za **nevhodné pro potřeby rychlého vývoje** v oblasti obrany. V roce 2019 prohlásil, že “*SAFe nepoužívá žádná úspěšná softwarová firma (Facebook, Google...), hlavním cílem vývoje je inovovat, ne být SAFE*” a zakázal jeho použití v projektech USAF s odůvodněním, že SAFe je “*antitezí*”

DevSecOps” (brání rychlému nasazování). Tak ostrý postoj naznačuje, že v daném kontextu SAFe zklamal očekávání natolik, že byl **úplně zavržen**.

- Některé velké korporace po počátečním nadšení SAFe opustily či výrazně zredukovaly. Neoficiálně se hovoří o případech, kdy firma po pár PI zjistila, že přínosy neodpovídají vynaloženému úsilí, a přešla na méně striktní model (např. vlastní variantu agile nebo jiný framework). Tyto *“tiché pivoty”* však nebývají publikovány.

Poučení: Úspěch SAFe **není zaručen** – rámec může pomoci, ale **negativní důsledky hrozí**, pokud se aplikuje mechanicky nebo v nevhodném prostředí. Organizace musejí zvážit svou kulturu a cíle: v některých kontextech může robustní rámec jako SAFe přinést pořádek a škálování, jinde může udusit kreativitu. Agilní guru Mike Cohn to shrnuje: *“žádný framework vám agilitu nezajistí, záleží na postojích lidí”*. Při posuzování SAFe je tedy vhodné studovat jak *success stories*, tak *lessons learned* z nezdarů, a podle toho přizpůsobit vlastní cestu.

Alternativní přístupy k agilnímu škálování

SAFe není jediný způsob, jak škálovat agilitu ve velké organizaci. Existují i jiné frameworky a modely, které si kladou za cíl **zvýšit zapojení zaměstnanců a dodat vyšší hodnotu zákazníkům** při zachování agilních principů. Mezi nejznámější alternativy patří **Large-Scale Scrum (LeSS)**, **Nexus** a tzv. **Spotify Model**. Každý z nich přistupuje k škálování odlišně:

Large-Scale Scrum (LeSS)

Large-Scale Scrum (LeSS) je rámec pro škálování Scrumu vytvořený Craigem Larmanem a Basilem Voddem (první kniha vyšla 2010). Jeho motto by mohlo znít: *“LeSS je prostě Scrum ve velkém”*. Autoři LeSS zdůrazňují, že většina osvědčených praktik pro malé týmy platí i ve velkém, a není třeba je nahrazovat novými procesy – spíše je třeba **odstranit organizační překážky**, aby mnoho týmů mohlo pracovat jako jeden organismus.

Hlavní **principy a přístupy LeSS**:

- **Minimalismus a zjednodušení:** LeSS záměrně udržuje **co nejjednodušší strukturu**. V ideálním případě až 8 Scrum týmů pracuje na jednom produktu **s jedním společným Product Backlogem a jedním Product Ownerem**. Nejsou zde žádné další “mezivrstvy” (žádný program manažer, RTE apod.). Všechny týmy plánují společně (*Overall Sprint Planning*), poté si rozdělí backlog položky a v rámci sprintu spolupracují, aby na konci dodaly **jednu společnou verzi produktu**. Organizační hierarchie je **co nejplošší** – LeSS přímo vyzývá firmy k *“de-scaling”*, tedy rušení složitých struktur, namísto přidávání dalších vrstev.
- **Zachování Scrum rolí:** LeSS nemá žádné nové role oproti Scrumu. **Product Owner** v LeSS spravuje backlog pro víc týmů najednou (musí tedy pracovat na vyšší úrovni abstrakce a delegovat detailní porozumění požadavkům na týmy). **Scrum Master** může působit pro více týmů (max 3), pokud to zvládne, a stará se o aplikaci LeSS pravidel. **Týmy** jsou plně kross-funkční a samostatné. Důraz je kladen na **kompetentnost týmů** – ideálně by měly zvládnout vše od analýzy po nasazení.
- **Celoproduktové zaměření:** Všechny týmy v LeSS spolupracují na **jednom produktu**. Není přípustné dělit týmy podle komponent či subsystémů – místo toho LeSS

preferuje **feature teams** (týmy schopné dodat funkčnost end-to-end). To zvyšuje **hodnotu pro zákazníka**, neboť každá sprint dodávka přináší ucelené funkce, nikoli jen dílčí moduly. Také to zvyšuje zapojení členů týmu: vidí přímou souvislost své práce s výsledným produktem, a často i se **zákazníkem**. (LeSS podporuje, aby týmy **komunikovaly přímo se zákazníky či uživateli** kvůli zpětné vazbě, místo abychom vše filtrovali přes Product Ownera.)

- **LeSS Huge:** Pro opravdu velké produkty (více než 8 týmů) LeSS zavádí pojem **Requirement Area** – de facto menší produkt v rámci produktu. Týmy se rozdělí do těchto oblastí a pro každou oblast je Area Product Owner, který spolupracuje s hlavním Chief Product Ownerem. I tak se ale snaží zachovat **co nejméně vrstev** – pořád platí jedna společná dodávka produktu. LeSS Huge tedy škáluje do stovek vývojářů, ale stále aplikuje principy jako jeden backlog a co nejvíc plochá struktura.

Angažovanost a kultura: LeSS svým důrazem na *opravdovou samoorganizaci* klade velkou odpovědnost i pravomoci na týmy. Týmy mají **svobodu způsobu práce** (metody inženýrství, detailní proces uvnitř sprintu) za podmínky, že dodržují pravidla nutná pro koordinaci s ostatními (společné Plánování 1, účast na společném Sprint Review atd.). To může vést k **vyššímu engagementu** – lidé se necítí svázáni korporátní byrokracií a mohou se soustředit na dodávání hodnoty. Na druhou stranu LeSS často vyžaduje **výraznou organizační změnu** (např. rozpustit projektové manažery, přeskládat týmy kolem produktů místo komponent). To je náročné a může vyvolat odpor uvnitř firmy. Není tedy náhodou, že LeSS se prosadil méně než SAFe – jeho *čistě agilní* filozofie je krásná, ale obtížně dosažitelná tam, kde kultura zatím není agilní. Přesto firmy, které LeSS úspěšně aplikovaly (např. BMW, UBS, Ericsson), hlásí **zvýšenou pružnost** a spokojenost týmů díky odstranění zbytečných mezičlánků.

Nexus

Nexus je *lehkou* škálovací nadstavbou Scrumu vyvinutou Kenem Schwabem a Scrum.org (představen kolem 2015). Cílí na škálování Scrumu pro **3 až 9 týmů** pracujících na jednom produktu. Název znamená „spojnice“ – Nexus má pomoci týmům udržet si jednotný směr a integrovat výsledky každého sprintu.

Hlavní **prvky Nexus:**

- **Jednotný backlog a Product Owner:** Stejně jako LeSS, i Nexus trvá na tom, že pro více týmů musí být **jeden Product Backlog a jeden Product Owner**. Všechny týmy tedy sdílejí *co* se bude dělat, liší se jen *jak to rozdělí mezi sebe*. Tím je zajištěno, že se pracuje opravdu na jednom produktu a podle jedné priority.
- **Nexus Integration Team:** Nexus přidává **integrační tým** – malý tým (či spíše role), který dohlíží na to, že všechny dílčí práce od týmů se integrují do jednoho funkčního celku. Tento tým typicky zahrnuje Product Ownera, možná pár senior vývojářů či architektů a zástupce jednotlivých Scrum týmů. Není to *manažerská* vrstva, spíše **servisní tým**, který pomáhá ostatním odstraňovat závislosti a integrační problémy. Prakticky třeba nastavuje CI/CD pipeline, koordinuje integrační testy apod., aby na konci sprintu vznikl **shippable Inkrement** za všechny týmy.
- **Události v Nexus:** Kromě standardních Scrum událostí Nexus definuje extra synchronizační ceremonie:

- **Nexus Sprint Planning:** zástupci všech týmů (např. Scrum Masteri, PO a klíčoví členové) se sejdou před začátkem sprintu, aby sladili obsah sprintu mezi týmy. Rozdělí backlog položky, proberou závislosti. Následně týmy dokončí své detailní plánování (Sprint Planning část 2).
- **Nexus Daily Scrum:** Každý den proběhne krátké setkání zástupců týmů (např. Scrum Masteri), kde se řeší **problémy přesahující jeden tým** – typicky integrace, blokery, co je potřeba sladit. Není to náhrada klasických Daily Scrum v týmech, ty probíhají také.
- **Nexus Sprint Review & Retrospective:** Na konci sprintu se výsledný integrální produkt prezentuje společně (jedno Review) a proběhne i **společná retrospektiva Nexus** kromě týmových retrospektiv. Ta se zaměřuje na zlepšení *škálovaného procesu* – např. jak lépe zvládat závislosti.
- **Minimální změny Scrumu:** Nexus se snaží **neměnit Scrum, jen ho rozšířit** pro více týmů. Týmy stále mají své Scrum Mastery (často jedna osoba může dělat SM pro 2–3 týmy). Důraz je na technických praktikách jako **Continuous Integration** – aby se předešlo integračním problémům. V tomto ohledu Nexus klade na týmy větší nároky v disciplíně: průběžná integrace kódu, testování napříč týmy atd. je nutností, ne volbou.

Princip: Ken Schwaber říká, že Nexus “**jde k jádru škálování minimalizací závislostí a integračních potíží**”. Místo aby přidával vrstvu řízení, snaží se *technicky* zajistit, že i deset týmů dokáže dodat hotový produkt každé 2–4 týdny. To má přímý dopad na **hodnotu pro zákazníka** – integrace je průběžná, takže zákazník nečeká, až se dílčí výstupy různých týmů složitě spojí někdy po měsících.

Engagement: Nexus, podobně jako LeSS, drží agilní princip “*jeden tým*” i pro větší celek. Týmy vidí, že přispívají k **jedné společné vizi produktu**, a sdílejí odpovědnost za výsledný increment. To může zvýšit jejich pocit smyslu – nejde jen o “můj modul”, ale o úspěch celého produktu. Zároveň díky omezení byrokracie (Nexus přidává opravdu minimum rolí) se vývojáři necítí svázaní procesem navíc. Mohou se soustředit na řemeslo, spolupracovat napříč týmy (typicky formou *Communities of Practice* či ad-hoc synchronizace na Nexus Daily). Nexus je tedy hodně blízký původnímu Scrumu – vyžaduje ale **vyzrálé Scrum týmy** a zkušeného Product Ownera, jinak snadno narazí (Nexus sám o sobě problémy nevyřeší, jen je činí viditelnými).

Spotify Model

Spotify model není formální framework, ale spíše **případová studie organizačního uspořádání**, které si vyvinula společnost Spotify kolem roku 2012. Proslavily ho články a videa od Spotify koučů Henrika Kniberga a Anderse Ivarssona (“*Scaling Agile @ Spotify*”). Mnoho firem se tímto modelem inspirovalo, protože ukazuje, jak může velká organizace zůstat **inovativní, flexibilní a atraktivní pro zaměstnance**.

Hlavní **prvky Spotify modelu**:

- **Squads, Tribes, Chapters, Guilds:** Základní jednotkou jsou **Squads** – malé *autonomní týmy* (5–12 lidí) podobné Scrum týmům. Každý squad je odpovědný za nějakou oblast

funkcionality (např. vyhledávání v aplikaci). Některé squads jedou Scrum, jiné Kanban – Spotify nechává na týmu *zvolit si proces*, důležitý je výsledek. Squady, které mají blízkou doménu, jsou seskupeny do **Tribes** (kmenů). Tribe je něco jako větší oddělení (do ~100 lidí) – např. tribe “Doporučování obsahu” může mít 8 squadů, každý řeší jiný aspekt. Tribe má **Tribe Leadera** (manažera), který se stará o provoz kmene, ale squads uvnitř mají velkou volnost. Aby se zabránilo izolaci týmů a sdílely se znalosti, existují **Chapters** – horizontální *kompetenční skupiny* napříč squads v jednom kmeni. Například všichni test inženýři v tribe tvoří Testing Chapter, vedou společné schůzky, sdílejí postupy. Chapter má **Chapter Leadera** – to je linový manažer pro členy chapteru (řeší jejich rozvoj, mzdy apod.), zároveň je *ale součástí jednoho ze squadů*, takže je „u toho“. Chapter Leader tak není byrokrat, ale spíše senior odborník-coach. Dále existují **Guilds** – *volná společenství* napříč celou firmou, kam se lidé sdružují podle zájmu (např. Front-end Guild může zahrnovat vývojáře ze všech tribe, kteří sdílejí know-how). Guildy nemají formální pravomoci, jde o komunitu praxe. Tato síť guild a chapter vytváří **knowledge sharing kulturu** napříč squads.

- **Autonomie a zodpovědnost:** Spotify model klade extrémní důraz na **autonomii týmů**. Motto bylo: “*Aligned Autonomy*” – tj. teams mají autonomii *co největší*, ale v rámci jasně sdílené vize (alignment). Každý squad má **misí** a může relativně samostatně rozhodovat *jak* jí dosáhnout. Pravidlo zní: „*pravidla jsou dobrý start, pak je poruš, když je třeba*“. Tedy žádný proces není posvátný, důležité je dodržet hodnoty. Jednou z hodnot je “**věřit lidem**” – vedení dává důvěru, že týmy učiní dobrá rozhodnutí. Např. squad může sám nasadit novou verzi služby do produkce, kdy uzná za vhodné (Spotify toho dosáhlo automatizací a architekturou mikroservis, aby týmy byly co nejvíc nezávislé). Tím pádem je **rychlost inovací vysoká** – žádné čekání na povolení shora.
- **Kultura před procesem:** V Spotify říkají, že skutečným základem je **inženýrská kultura** postavená na *motivaci, komunitě a důvěře, spíš než na struktuře a kontrole*. Firmu drží pohromadě společné **hodnoty**, ne byrokratické předpisy. Například hodnota “*užij si to*” podporuje, aby práce byla i zábava; hodnota “*být otevřený*” zas, že se problémy sdílí napříč. Díky tomu Spotify dosahovalo mimořádně **vysoké spokojenosti zaměstnanců (94 %)** – lidé oceňovali volnost a důvěru, kterou dostali, i možnost učit se od kolegů v chapterech a gildách. Tato kultura pak přímo vede k lepším produktům: angažovaný a motivovaný tým produkuje inovace a rychle reaguje na změny trhu.
- **Neustálá evoluce modelu:** Spotify neustále experimentovalo s tím, co funguje. Model nebyl statický – např. zavedli pojem “**Trio**” (trojice vedoucích role pro produktový směr – typicky produktový manažer, designér a tech lead spolu určují strategii pro tribe) nebo “**Alliance**” (volné spojení více tribes, pokud je produkt hodně velký – obdoba programu). Důležité je, že nic z toho nebylo *přikázáno zvenku*; vzniklo to organicky dle potřeb. Spotify model je tedy spíše *myšlenkový vzor* než šablona. I autoři upozorňovali, že **nejde o univerzální framework** – jiné firmy si musí najít vlastní cestu, poučit se z principů, ne slepě kopírovat strukturu.

Přínos pro zákazníky a zaměstnance: Spotify model prokazatelně vedl k rychlému růstu firmy – Spotify zvládlo škálovat na stovky vývojářů a vydávat nové funkce velmi rychle, čímž si získalo 10× více uživatelů během pár let. **Zákazník** profitoval z neustálých vylepšení aplikace, protože autonomní squads mohly kontinuálně dodávat (tzv. model “*tiny releases*” –

mnoho malých nasazení denně). **Zaměstnanci** zase oceňovali volnost – vznikla *silná kultura angažovanosti*, lidé se cítili být součástí komunity. Model kapitol a gild podporoval odborný růst (sdílení know-how) a přitom nikdo nemusel opouštět svůj tým. To vše vedlo k tomu, že Spotify bylo považováno za **atraktivního zaměstnavatele a inovativní firmu**.

Mnoho korporací (např. ING bank, Tesco, Rijkswaterstaat v Nizozemí) se pokusilo převzít Spotify model v rámci svých agilních transformací, často s cílem **zvýšit engagement zaměstnanců** a odbourat strnulé struktury. Úspěšnost těchto pokusů se liší – některé organizace dokázaly prvky modelu úspěšně adaptovat (např. ING reportovala vyšší spokojenost klientů i zaměstnanců po “Tribes” reorganizaci), jiné narazily, protože **převzaly formu bez kultury**. Hlavním poučením ze Spotify modelu je, že **hodnoty a principy (autonomie, důvěra, orientace na lidi)** by měly vést strukturu, nikoliv naopak. Pokud firma nejprve změní myšlení na všech úrovních, může si pak vytvořit vlastní “Spotify-like” uspořádání, které přinese jak větší radost z práce pro lidi, tak lepší produkty pro zákazníky.

Závěr

Škálování agility ve velkých organizacích je náročný podnik, a proto vznikají různé přístupy jako SAFe, LeSS, Nexus, Spotify model a další (např. Scrum@Scale, Disciplined Agile Delivery). **Scaled Agile Framework (SAFe)** představuje nejrozšířenější „učebnicový“ postup, který se snaží skloubit agilní a lean principy s potřebami korporací. Je však zřejmé, že **názory na SAFe se různí** – pro některé je to praktická cesta k agilitě v celopodnikovém měřítku, pro jiné popření agilních ideálů. Při porovnání s Agilním manifestem vidíme, že SAFe klade větší důraz na **procesní rámec a koordinaci**, zatímco manifest akcentuje **flexibilitu, jednoduchost a lidský faktor**. Klíčové je, *jakým způsobem* je SAFe implementován – zda s důrazem na mechanické dodržování, nebo se skutečným pochopením agilních hodnot.

Alternativní škálovací rámce jako **LeSS a Nexus** ukazují, že to jde i jinak: spíše **uberou manažerské řízení** a vsadí na *evoluční samoorganizaci* většího počtu týmů. To může vést k vyšší kreativitě a angažovanosti, ale vyžaduje to **vyzrálou agilní kulturu** a ochotu měnit zaběhlé struktury. **Spotify model** zase demonstruje sílu *principu „people over process“* – volná struktura postavená na autonomii a komunitě může fungovat a přinést skvělé výsledky, pokud je podepřena silnými společnými hodnotami.

Zvýšení engagementu zaměstnanců a dodání vyšší hodnoty zákazníkům jsou cíle, které spojují všechny tyto přístupy. Každý k nim však jde jinou cestou – od **přísně definovaného orchestru** (SAFe) po **improvizovaný jazz** (Spotify). V praxi možná neexistuje *jednoznačná odpověď*, který rámec je nejlepší; záleží na kontextu firmy, její kultuře, lidech a produktech. Důležité je neztratit ze zřetele podstatu agility: **schopnost rychle se učit a přizpůsobit, spolupracovat napříč týmy a maximálně se orientovat na přínos pro zákazníka**. Framework by měl těmto cílům sloužit, nikoli je brzdít.

Výše rozebrané zkušenosti – úspěchy i neúspěchy – poskytují cenné poznatky. Organizace usilující o agilní škálování by měly tyto poznatky využít a možná si *vytvořit svůj hybridní model*, šitý na míru jejich prostředí. Ať už s pomocí SAFe či jiného rámce, klíčové zůstává: **agilita začíná v hlavách a srdcích lidí, nikoli v procesních diagramech**. Pokud se podaří tuto agilní *mindset* udržet i ve velkém měřítku, pak jakýkoli zvolený přístup může vést k vyššímu

engagementu zaměstnanců **a skrze něj i k vyšší hodnotě dodané zákazníkům** – což je konec konců společným jmenovatelem Agilního manifestu i všech škálovacích frameworků.