

Sprint

Iterace jsou jedním ze základních principů, které agilní metody využívají. Tou nejznámější, která ve Scrumu dostala pojmenování, je Sprint. Vychází z pozorování, že pravidelně se opakující věci jsou pro lidi z psychologického hlediska příjemné a snadno si na ně zvyknou. Proto Scrum celý vývojový proces rozděluje na pravidelné cykly, Sprintsy, ve kterých tým dodává hotovou funkcionalitu. Vzhledem k pravidelnosti cyklů jsou výsledky prezentované pokaždé včas a vzhledem k procesu plánování dostaneme pokaždé to, co jsme čekali. Tedy po nějaké době, nic nejde hned, že.

Sprint je fixní časový úsek, který má tým na to, aby dosáhl cílů Sprintu (Sprint Goal). Jak dlouhý by měl takový Sprint být? Vzhledem k tomu, že na jeho konci je Sprint review pro zákazníky, kdy bychom jim měli prezentovat hotovou funkcionalitu, musí být alespoň tak dlouhý, abychom stihli jako tým něco dokončit, a měli tak co prezentovat. Na druhou stranu, příliš dlouhým Sprintem se okrádáme o častou možnost zpětné vazby, a to jak od zákazníka, tak interně od členů týmu.

Zároveň by měla délka Sprintu reflektovat dynamiku businessu. Každou iteraci se učíme, získáváme zpětnou vazbu od zákazníka a ověřujeme business hodnotu. Začátkem každého Sprintu můžeme změnit či upravit jak produkt, tak naše procesy a praktiky. Obecně platí

že čím kratší Sprint, tím lepší. Čím kratší Sprint, tím rychlejší zpětná vazba a adaptace na změnu.

Když celý tým spolupracuje vždy na jedné položce backlogu, Sprint vlastně nemůže skončit neúspěchem, protože tým dokončuje položky backlogu postupně.

Cílem Sprintu není dodat všechny položky Sprint Backlogu, ale dosáhnout cíle sprintu (Sprint Goal).

Jak nastavit Sprintsy:

- Nevíte-li, jak dlouhý Sprint zvolit, začněte dvěma týdny. Delší Sprintsy v podstatě nemají smysl, protože nedávají dostatečně častou zpětnou vazbu.
- Máte-li pocit, že za Sprint nestihnete nic dokončit, pak než Sprint prodloužíte, zamyslete se nad tím, proč tomu tak je. Nesnažte se dokončit příliš velké kusy funkcionality najednou? Nedá se něco vylepšit na spolupráci v rámci týmu?
- To, že jste nestihli dokončit všechny položky Sprint Backlogu, není důvod pro to, aby se Sprint změnila délka. Ta zůstává stejná, výsledek zohledníte v plánu příštího Sprintu a při retrospektivě.
- Cíl Sprintu (Sprint Goal) by se neměl během Sprintu měnit, je to důvod, proč do něj investujeme. Je na týmu, aby se v rámci Sprintu zorganizoval a pro dodání cíle Sprintu udělal maximum.

Výsledky následně prezentuje na Sprint Review. Finální výsledky tedy není možné od týmu požadovat v průběhu Sprintu.

Sprint Backlog se v průběhu Sprintu změnit může, je to jen plán, jak bychom mohli cílů Sprintu dosáhnout.

Souvislosti:

(co budete ještě potřebovat zavést)

Sprint Goal: Je klíčovým prvkem Scrumu. Je takovou krátkodobou vizí, definuje, jaké business hodnoty chceme v průběhu Sprintu dosáhnout.

Sprint Backlog: Vychází z Product Backlogu, a je to plán, jak z business pohledu chceme dosáhnout cíle Sprintu.

Planning: V rámci Planning meetingu tým naplňuje Sprint Goal, Sprint Backlog a domluví se, jak na tom budou spolupracovat.

Tabule: Průběh Sprintu je nejlépe vidět na tabuli. Nepotřebujete k tomu žádné složité nástroje. Přehledná tabule na zdi funguje mnohem lépe a nepřináší zbytečný overhead.

Standup Meeting: Abyste si byli na denní bázi jisti, že ještě pořád stihnete jako tým dokončit to, k čemu jste se zavázali.

Sprint Review: Tým prezentuje, jak dosáhl cílů Sprintu, aby na dodanou business hodnotu získal zpětnou vazbu od zákazníků.

Product Backlog

em projektu si někam zaznamenáváte, co plánujete udělat. Někdy je tento seznam i přímo softwarovým týmem, někdy produkt y, jindy přímo zákazníkem. Ve Scrumu takový nazýváme Product Backlog a dáváme ho stí Product Ownerovi, ale měl by být otevřený ný všem. Jednotlivé položky Product Backlogu odpovídá jednotlivým funkcionalitám pro dy něčemu, co přináší zákazníkovi hodnotu.

Owner samozřejmě není na sestavení a udržo- duct Backlogu sám, pomáhá mu Development lakeholderi. Product Owner má poslední slovo povědný za priority a celkovou smysluplnost ků pro zákazníka.

Backlog obvykle obsahuje nejen jednotlivé ale i odhad complexity/náročnosti v relativ- notkách ohodnocený týmem a prioritou. Prio- xcionalit jsou rozpracované do většího detailu, ioritní celky mohou zůstat klidně na úrovni vel- ců.

Backlog by se dal přirovnat k velkému afric- onovi. A cílem vaší restaurace není celého slona arpaccio a pak ho začít teprve podávat. Carpac- áte jen z malého kousku, z trochu většího kusu připravíte menší kostky a zbytek necháte zatím

v celku. A jak jde čas, doplňujete zásoby kostek i carpa- ccia. S Product Backlogem se to má podobně. Zrovna tak jako maso by mohlo oschnout a špatně se sklado- vat, Backlog se může měnit. Ať už protože nastaly nové okolnosti, nebo jako důsledek zpětné vazby od zákaz- níka. Ostatně, kolik z vás si až příliš často stěžuje, že zákazník vlastně neví, co chce. A proto by dopodrobna připravené analýzy plné detailů mohly časem přijít vni- več.

Správný Product Backlog je seřazený podle priority do pyramidy. Na jejím vrcholu jsou ty nejdůležitější User Stories, kvalitní, takové, aby je tým byl schopen akceptovat do Sprint Backlogu a dokončit v rámci Sprintu. Nezávislé, jasně popsané funkčnosti, o kterých víme, jak se pozná, že je User Story hotová a čeho má dosáhnout. Carpaccio, když se vrátíme k analogii se slo- nem. Takových plátků carpaccia ve formě User Stories by mělo být v každý okamžik připraveno na vrcholku pyramidy na přibližně 2 až 3 Sprints práce dopředu.

Backlog se sestává z funkčních celků, které přináší zákazníkovi hodnotu.

Technické úkoly (jako například refaktoring) do Backlogu nepatří.

Hned pod nimi v pyramidě je sekce stále ještě hodně prioritní práce, rozdělené na menší kusy, které ještě nemusí být do detailu definované, ale měli bychom mít

už o takové funkcionalitě docela konkrétní představu. Takových připravených kostek je v Product Backlogu přibližně na dalších 5 až 10 Sprintů.

Zbytek Backlogu může být klidně hodně nejasný, pouze ve formě velkých Epiců, ostatně nejedná se o prioritní práci a pravděpodobnost, že se funkcionalita ještě změní, je tak hodně vysoká. Samozřejmě můžete mít rozpracované i méně prioritní věci, ale jestliže chcete jako Product Owner něco dát na úpinou špičku pyramidy a nechat to týmem dokončit v následujícím Sprintu, musíte už velice dobře vědět nejen to, co chcete, ale i pro koho a proč, ale hlavně to, jak se pozná, že je daná User Story hotová.

Backlog se tedy sestává z položek backlogu, které přináší hodnotu, obvykle ve formě User Story. Pro předstihu uvádíme, co by takový řádek Product Backlogu mohl obsahovat za položky:

- #ID**
- Epic** (reference/jméno pro snadnou orientaci a filtrování)
- Theme** (reference/jméno pro snadnou orientaci a filtrování)
- Jméno US** (krátké jméno pro snadné zapamatování)
- US** (Jako uživatel chci funkcionalitu, abych dostal BV)

Review Comments (kde obvykle tým zaloguje informace o review)

Test (kde tým loguje informaci o testování které testy prošly a kdo je použítel)

Attachment (např. obrázek, jak to má vypadat)

Comments (pro cokoli, co si potřebujete poznamenat)

Estimates

Priority

Na úrovni Product Backlogu nás zajímají hlavně User Stories. Reference na větší celky je obecně doporučované udržovat ve třech úrovních: Epic, Theme a User Story, kde Epic odpovídá větší položce backlogu, která se ještě musí rozdělit na menší User Story, Theme sdružuje nezávislé položky Backlogu, které spolu logicky souvisí, a User Story jsou položky dostatečně jasně a malé, připravené pro naplánování do Sprintu. Je vhodné mít takový Backlog někde uložený, abyste viděli historii a mohli se kdykoli podívat, co v něm je. Nejsnazším nástrojem je Excel nebo Google Sheets. Funguje a je flexibilní.

Složitějších nástrojů je mnoho. Ale většinou kromě očekávaných výhod přinesou i velký overhead a zbytečnou administrativu spojenou s komplexitou nástroje, tak vybírejte opatrně. Product Backlog obvykle neobsahuje tasky, ač to některé nástroje nabízejí. Je to zbytečná úroveň detailu, kterou si jednotlivé týmy vedou

na svých Scrum tabulích a nástroj na ni nepotřebují. Na závěr je dobré jednotlivé commity prolinkovat s danou US, abyste i v budoucnu věděli, kdo které změny dělal a proč.

Souvislosti:

(co budete ještě potřebovat zavést)

Product Owner: Je zodpovědný za prioritizaci Product Backlogu.

Product Backlog Item: Product Backlog je prioritizovaný list takzvaných Product Backlog Items, obvykle se používá forma User Story.

Sprint Backlog: Obsahuje výběr z Product Backlogu do konkrétního Sprintu.

Sprint Backlog

Sprint Backlog je jen malou součástí Product Backlogu a obsahuje ty prioritní funkcionality, které tým plánuje dodat v rámci Sprintu. Sprint Backlog vybírá tým podle priorit Product Ownera tak, aby jim pomohli dosáhnout Sprint Goalu, na začátku každého Sprintu. Tým si obvykle v průběhu Sprintu jednotlivé funkcionality rozpadne na konkrétní úlohy, které pak v průběhu Sprintu zpracovává, a má tak větší přehled o rizicích a statusu vzhledem k dodání celého závazku ze Sprint Backlogu,

na svých Scrum tabulích a nástroj na ni nepotřebují. Na závěr je dobré jednotlivé commity prolínkovat s danou US, abyste i v budoucnu věděli, kdo které změny dělal a proč.

Souvislosti:

(co budete ještě potřebovat zavést)

Product Owner: Je zodpovědný za prioritizaci Product Backlogu.

Product Backlog Item: Product Backlog je prioritizovaný list takzvaných Product Backlog Items, obvykle se používá forma User Story.

Sprint Backlog: Obsahuje výběr z Product Backlogu do konkrétního Sprintu.

Sprint Backlog

Sprint Backlog je jen malou součástí Product Backlogu a obsahuje ty prioritní funkcionality, které tým plánuje dodat v rámci Sprintu. Sprint Backlog vybírá tým podle priorit Product Ownera tak, aby jim pomohli dosáhnout Sprint Goalu, na začátku každého Sprintu. Tým si obvykle v průběhu Sprintu jednotlivé funkcionality rozpadne na konkrétní úlohy, které pak v průběhu Sprintu zpracovává, a má tak větší přehled o rizicích a statusu vzhledem k dodání celého závazku ze Sprint Backlogu,

nim méně tyto úlohy jsou pouze interní praktikou týmu a nejsou součástí Sprint Backlogu.

Souvislosti:

(co budete ještě potřebovat zavést)

Product Owner: Je to klíčová role, člověk, který je za Product Backlog zodpovědný a je schopen určit priority.

User Story: Vhodná forma, jak popisovat jednotlivé položky Backlogu. Je stručná, ale postačující.

Ohodnocení: Jestli chcete alespoň tušit, kdy to, co jste si napsali do Product Backlogu, bude hotové, ohodnotte jednotlivé User Story. V agilním světě se obvykle ohodnocuje v relativních bodech.

User Story

Agilních týmech se často používá pojem User Story. Kládní formát User Story je:

„Jako *Uživatel* chci *Funkcionalitu*, abych dostal *Business Value*“

Kdo zapsaná User Story vám říká nejen to, co chcete dostat, ale i pro koho, a hlavně proč. User Story má tvářet obrázek, popisuje příběh. Lidský mozek vnímá rázky a příběhy daleko snadněji než technický popis pododch. Zkuste si teď porovnat jeden příklad z praxe:

US: „Kontaktní formulář“ (jaký jste si představili?)

US: „Jako administrátor chci kontaktní formulář, bych se včas dozvěděl o chybách systému.“

velice pravděpodobně, že když jste to četli, vybavily vám dvě různé věci. Myslíte si, že je to z kontextu my a produktu jasné? Ne vždy. Většinou je to jasné pouze Product Ownerům, ti mají v hlavách obrázky, jak esně se má systém chovat a jak má vypadat. Ti jsou učástí příběhu zákazníka, ale v User Stories jde o to, vy ten obrázek, co mají v hlavě, sdělili ostatním tak, y ho ani oni nikdy nezapomněli. A aby se i tým stal učástí příběhu zákazníka.

User Story by měla být jednoznačně popsateľná, tvářet obrázek, být nezávislá, přinášet hodnotu,

a také malá – v originále je to Independent, Negotiable, Valuable, Estimable, Small, Testable (zkráceně „INVEST“). Tedy něco, do čeho chcete investovat úsilí a potažmo peníze. Je-li User Story příliš velká, je většinou těžké říct, co je jejím obsahem a co už ne, a je tak pro tým neuchopitelná. Diskuse o ní se vlečou a Product Owner dokola vysvětluje, co že to je.

User Story popisuje funkcionalitu. Ne tu existující, ale tu, kterou chcete přidat nebo změnit. Není to detailní specifikace ani seznam technických aspektů, User Story je vždy zaměřená na business value a popisuje přínos, který od ní očekáváme. Jako taková dodává v ideálním případě end to end funkcionalitu, kterou můžete rovnou použít nebo na ni získat zpětnou vazbu.

A jak se pozná, že je User Story hotová? Adresovala danou hodnotu. Jak přesně ji tým naimplementuje, to je na dohodě Scrum týmu v rámci Sprintu. User Story můžete samozřejmě dělit na menší User Story, pořád ale musí přinášet hodnotu. A hodnotou nejsou technické aspekty problému ani architektura systému, ani specifikace. Na User Story se díváme z pohledu businessu, z pohledu uživatele. Technologie přichází na řadu až při rozpadu User Stories na jednotlivé úlohy.

Období vizualizace

liní metody už z pohledu prvního bodu Agilního manifestu dávají přednost „jednotlivcům a interakci s procesy a nástroji“. Agilní metody ale často ve firmách zavádějí softwaroví inženýři a ti mají nástroje moc. Až příliš často považují za klíčovou otázku, který nástroj budeme na podporu našeho Scrum procesu / procesu používat. Čím složitější, tím lépe. Pojdme se podívat, co takový tým opravdu potřebuje.

Uhlavní tabule je klíčem k efektivnímu sdílení informací v rámci týmu.

U tabule by mělo být vidět, jaké User Stories má tým dokončit a jaký je jejich status, tedy co zbývá dokončit a je již hotové, kdo na čem pracuje.

Uhlavního pohledu potom, jestli tým ještě pořád věří, dosáhne cíle Sprintu.

Uhlavní je to jen trochu možné, dejte alespoň ze začátku pozornost fyzické tabuli před elektronickými nástroji.

Uhlavní opřekvapivě, ale stačí obyčejná tabule nebo zeď, kam můžete lepit lístečky. Snadné. Obejde se to bez nákupu speciálních nástrojů i nastavování. Je to flexibilní, kdykoli ji můžete přemístit a začít s ní pracovat jinak. Nijak vás ani váš proces neomezuje. A hlavně je po ruce. Nemusíte nikam klíčenku přenášet, prostě hned z místnosti, kde celý tým sedí, vidíte, jaké úkoly jednotlivé úkoly jsou a kdo na čem pracuje.

Scrum Tabule

Jak taková tabule vypadá? Obecně stačí, když má tři sloupce: Sprint Backlog, In Progress, Done. Na začátku Sprintu se všechny naplánované User Stories dají do sloupečku Sprint Backlog a v ideálním případě se rozpadnou na jednotlivé úkoly.

V průběhu Sprintu se potom každý den jednotliví členové týmu zavazují k dokončení konkrétních úloh s tím, že se snaží minimalizovat rozpracované úkoly a dokončovat naplánované User Stories. Obvykle se tým domluví, že každý člen týmu smí mít pouze jednu rozpracovanou úlohu v daný čas, nicméně více členů týmu může v jeden čas spolupracovat na jedné úloze.

Než začnete pracovat na další úloze, musíte úlohu dokončit nebo někomu předat. Každý člen týmu tak může mít nějakou unikátní značku, kterou rozpracované úkoly označuje. Z tabule je pak krásně vidět, v jakém stavu která User Story je a kdo na čem pracuje.

Souvislosti:

(co budete ještě potřebovat zavést)

Sprint: Práce v iteracích je dobrá pro větší předvídatelnost, jak na tom jako tým jsme.

Sprint Backlog: Abychom mohli sledovat závazek týmu na daný Sprint, je dobré před každým Sprintem zafixovat plán.

Tým: Když tabule zobrazuje jen individuální status jednotlivých lidí, obvykle ji nikdo neupravuje a stává se z ní jen nástroj reportingu.

Kanban Tabule

Kanban tabule musí splňovat tři základní principy Kanbanu, tedy:

- ☐ **Popište svůj proces a vizualizujte ho** – Kanban tabuli si můžete představit jako výrobní linku. Tabule mívají více sloupců, v každém sloupci je jiná činnost, na které obecně pracuje jiný tým lidí. Nenechte nijak omezovat svou fantazii v možnostech vizualizace stavu konkrétní úlohy. Kartičku označte avatarem, ať každý na první pohled vidí, kdo na čem pracuje. Tečkou můžete označit dny, kdy se na ní pracovalo, natržením dny, kdy se čekalo na něho zvenku. Možností je nekonečně mnoho. Je to kreativní proces,

zkoušejte a měňte ho. Princip vizualizace týmy používají i pro Scrum tabule.

- ☐ **Omezte rozpracovanou práci** – Tady je Kanban poměrně striktní a zavádí v každém sloupci limitované fronty. Obvykle si týmy na tabuli nakreslí prázdná pole, kam je možné kartičku posunout. Jsou-li všechna pole plná, musí tým nejprve nějakou aktivitu dokončit, aby mohl začít další.

- ☐ **Optimalizujte průchod systému** – Z takto navržené tabule je přímo vidět, kde proces není optimální. Podívejte se, kde jsou ve vaší výrobní lince díry, tedy kdy tým potřebuje posílit, případně kde tým čeká, tedy nemůže pracovat. Část vyřešíte limity jednotlivých stavů, část optimalizací procesu či počtu lidí.

Kdy potřebujeme nástroj?

V dnešní době samozřejmě existují i nástroje, které se snaží tabuli v elektronické podobě simulovat. Nicméně zážitek, který týmu poskytuje práce s fyzickou reprezentací takový nástroj nikdy nenahradí. Jeho dostupnost je výrazně nižší – musíte kliknout na odkaz a systémem otevřít a to většina členů týmu neudělá, dokud nemusí.

Pocit většího zapojení, kdy na tabuli vidíte kartičky psané svou vlastní rukou, se elektronickým nástrojem nedá nahradit. Navíc všechny změny, které se na tabuli

stanou, prostě vidíte, protože sedíte ve stejné místnosti a všimnete si, že někdo přišel k tabuli a něco přesunul. A i taková drobná změna, kterou vnímáte jen okrajově, funguje a přináší dobrý pocit z další dokončené práce.

Kdy tedy nástroj potřebujete? Prvním případem jsou distribuované týmy. Není-li možné posadit tým do jedné místnosti nebo je alespoň každý den svolat na Scrum meeting, asi budete potřebovat nějak sdílet data i v rámci odlišných geografických lokací.

Druhý případ je v podstatě platný pro všechny týmy a doporučuje nepoužívat nástroj ani pro plánování Sprintu, ani pro aktuální status Sprintu, ale jen jako archiv a repositář změn. Aby někde bylo zaznamenáno, kdo a kdy na čem pracoval a proč. Číslo záznamu je

obvykle svázáno se změnami ve verzovacím systému. Pro plánování Sprintu a jeho status se i nadále využívá fyzické tabule.

Když začínáte, bohatě vám pro trackování Backlogu postačí MS Excel nebo Google Sheet. Je flexibilní a nevynucuje žádný konkrétní proces. Po nějaké době, až si agilní procesy vyzkoušíte, můžete použít některý z mnoha nástrojů na trhu, dejte si ale pozor, aby rozsáhlé možnosti nástroje po chvíli nezačaly určovat, jak pracujete.

Nástroje jsou užitečnou pomůckou, ale neměly by nikdy určovat, ovlivňovat ani svazovat váš proces.

Hodnocení

jiní metody používají na ohodnocení User Stories obvykle relativní jednotky. Ohodnocuje vždy celý tým. Důvodů pro to je hned několik. Tím nejdůležitějším je, že když hodnotí více lidí, je menší pravděpodobnost, že něco neuvědomíte nebo se na funkcionalitu budete dívat například jen z úhlu vývoje a už ne z pohledu uživatele, a část práce tak podceníte.

Ohodnocujeme náročnost, komplexitu. Není to o čase. Hodnocení jsou relativní vůči sobě navzájem, jde tedy o kterýkoli moment říct, která User Story je větší, a tedy náročnější a která menší. Tým zohlední při ohodnocení nejen to, jak náročné je danou User Story naplnit, ale i otestovat a zdokumentovat. Prostě započítá všechny činnosti, které se musí dokončit, než bude User Story připravená na akceptaci zákazníkem.

Při ohodnocování zohledňujeme i míru rizika, tedy User Story, o které máme jen matnou představu, bude mít pravděpodobně vyšší hodnocení, někdy až nekonečno. Je pak indikací pro Scrum tým, aby ji společně rozdělil na menší a lépe popsatelné User Stories.

Přidávat něco můžete až tehdy, když tomu rozumíte. Nejjednodušší je User Story přiřadit kategorii. Funguje stejně, jako když jdete do obchodu pro tričko. Taký si můžete představit, co se stane, když si koupíte XS, S, M, nebo XL. XS vám bude malé, XL na vás bude povět-

šinou viset. S User Stories je to podobné. Na začátek si zvolíte nějakou referenční hodnotu a k ní pak ohodnocujete další User Stories. Každé další hodnocení pak je relativní vůči ostatním, již ohodnoceným funkcionalitám.

Tato stupnice se používá obvykle pro úvodní odhady, kdy ještě nemáme přesnou představu o konkrétních User Stories a známe jen velké Epiky a potřebujeme si udělat představu o rámcových nákladech/času. Když už jsme v projektu dál a chystáme se začít, měli bychom mít lepší představu o funkcionalitě, a být tedy schopni ji ohodnotit přesněji. Na to se používají body a tzv. planning poker. Stejně jako u velikosti triček se držíme relativního hodnocení a kategorií, jen jich je tentokrát více.

Dalo by se říct, že samotné ohodnocení je vlastně jen vedlejší produkt diskuse, kterou tým vede s Product Ownerem a mezi sebou. V momentě, kdy User Story rozumí, už je snadné ji ohodnotit. Nevíme-li ale, co vlastně zákazník chce, je i schopnost dát dané funkcionalitě nějaké ohodnocení relativně nízká.

Tým hodnotí pouze User Story. Hodnocení na úrovni jednotlivých technických úkolů nemá smysl. Je zdlouhavé a nemá žádnou přidanou hodnotu. V případě, že je hodnocení vysoké, je třeba User Story rozdělit na více menších User Stories.

Rychlost

Po skončení každého Sprintu tým sečte ohodnocení dokončených User Stories. Výsledek je jejich rychlost (velocity). Nedokončené User Stories v jakémkoli stavu rozpracování jdou zpět do Product Backlogu, a do rychlosti se tedy nepočítají. Dobře fungující týmy mají obvykle rychlost stabilní/předvídatelnou. Nedaří-li se vám to ze začátku takto dodržovat, pak udělejte-li si průměr rychlosti za poslední 3 Sprints, bude s největší pravděpodobností předvídatelnou i u začínajících týmů. Takto naměřená rychlost je pak číslo, se kterým můžete počítat každý Sprint.

V případě, že vám rychlost příliš osciluje, podívejte se na granularitu User Stories ve Sprint Backlogu. Na jeden Sprint byste měli mít naplánováno minimálně 5–10 User Stories. Největší User Story nesmí týmu trvat déle než polovina Sprintu. Zároveň je dobré omezit rozpracovanou práci a nezačínat žádnou User Story, kterou již nestihnete dokončit.

Rychlost týmu zjistíte až po čase. Na začátku ji samozřejmě můžete odhadnout, ale buďte vždy připraveni svůj odhad zrevidovat. Rychlost není žádné dogma. To, že tým „neplní“ očekávanou rychlost, má určitě nějaký důvod. Pravděpodobně pracuje více na chybách, technickém dluhu, učí se nebo jen má vaše firma vyšší overhead, než jsme čekali.

Rychlost je jedním z ukazatelů zdravoti týmu.

Dobře fungující tým má rychlost stabilní.

Tým je tak předvídatelný.

Tým, ve kterém má rychlost velké výkyvy, obvykle dostatečně nespolupracuje.

Rychlost vám ukazuje realitu. Takovou, jaká je. Když z ní uděláte tvrdou metriku, tým se nějak přizpůsobí a začne např. ohodnocovat vyššími čísly a vy jste ztratili základní výhodu tohoto přístupu. Rychlost měříme vždy na tým a Sprint, nikdy za jednotlivé členy. Podporujete tým spolupráci a zaměření na týmový výsledek před individuálním.

Začínáte-li nový projekt nebo máte nový tým, odhad rychlosti bude pravděpodobně jen orientační. Doporučuje se počítat s pozvolným nárůstem rychlosti po první tři Sprints, např: 5, 13, 26 při očekávané rychlosti kolem 30. Má to tu výhodu, že tým má dostatek času se seznámit s problematikou, ale zároveň už by měl i něco málo dokončit.

Nikdy nedělejte z rychlosti tvrdou metriku – tým se přizpůsobí (třeba tím, že zvýší odhady), jen aby splnil metriku, ale více práce neudělá.

Rychlost měřte vždy za tým a Sprint, nikdy za jednotlivce.

Meetingy a aktivity

Standup / Daily Scrum meeting

Jednou z nejznámějších praktik, které se v agilních metodikách využívají, je Daily Scrum meeting. Říká se mu Standup, případně denní Scrum. Tento meeting je poměrně snadný na implementaci, takže s ním většina týmů začíná. Teorie říká, že tým se sejde pravidelně každý den (obvykle ráno) u Scrum tabule či na jiném místě tak, aby jeho členové neseseděli na svých místech u počítačů, ale stáli – odtud se bere název standup – a sdílí informace o tom, na čem pracovali včera, na čem budou pracovat dnes a jestli mají se svým úkolem nějaké problémy, o kterých by tým měl vědět.

Meeting je týmovou záležitostí, tedy jednotliví členové si sdělují informace vzájemně. Role Scrum Mastera je čistě v pozici moderátora a facilitátora diskuse. Chce-li se kdokoli přijít podívat, je jistě vítán, ale jen v roli pozorovatele, který do dění nezasahuje a nesnaží se ovlivňovat průběh. Toto je obzvlášť těžké pro manažery, kteří mají rádi vše pod kontrolou.

Jestli vám Standup meeting trvá déle než 5 minut, pravděpodobně se nedržíte doporučeného formátu a utíkáte k technickým detailům.

Jak již bylo řečeno, Scrum meeting je dobrou platformou pro sdílení informací a zkušeností mezi jednotlivými členy týmu. Ale to není vše. Jeho primární význam je v posílení týmového závazku, tedy pocitu, že stále ještě věříme, že jako tým stihneme dosáhnout cíle Sprintu (Sprint Goal). Jak toho docílíme? Ideální je, když se Scrum meeting koná u fyzické tabule, kde je na první pohled vidět, které User Stories jsou ještě ve Sprint Backlogu, na kterých se již začalo pracovat, a také co už je hotovo.

Další výhodou je, když tým už v průběhu planningu rozpadne jednotlivé User Story na tasky, a tedy má každý člen týmu přehled, co nám, jako týmu, chybí k dokončení které konkrétní User Story. To, že tým stojí, také není nijak náhodně zvolené, výrazně to urychlí průběh meetingu. A teď už jen zbývá držet se doporučeného formátu a nechat každého člena týmu odpovědět na následující otázky:

- co jsem dokončil včera, co **pomohlo týmu dosáhnout cíle Sprintu**
- co dokoncím dnes co **pomůže týmu dosáhnout cíle Sprintu**
- problémy, které nám **stojí v cestě k cíli Sprintu**

Pokud jsme v situaci, kdy máme týmy v různých časových zónách, nebo nelze meeting dělat ráno, otázky si pozměníme, třeba na:

- co jsem **dokončil** od včerejšího standup meetingu
- co dokončím do dalšího standup meetingu
- co nám brání dosáhnout cíle Sprintu

Když to zvládnete do 15 minut a nezabřednete do žádných zdlouhavých diskusí, máte vyhráno. V praxi vám na tým o 7 lidech obvykle stačí 5 minut.

Ač je Standup na první pohled jednoduchý formát, musíme zdůraznit pár klíčových věcí. Nejdůležitější na celém Scrum meetingu je udržet pozornost jednotlivých členů týmu. Ostatně oni jsou hlavními zákazníky tohoto meetingu, a měl by tedy pro ně být zajímavý. Když se vrátíme k formátu výše zmíněných třech otázek, nejde jen o to si vzájemně říct, co jsme včera dělali. To obvykle nikoho zas tolik nezajímá. Vzhledem k závazku týmu dosáhnout cíle Sprintu je daleko důležitější mluvit o tom, co jsme opravdu dokončili a pomohlo nám cíle Sprintu dosáhnout. A zpět k tabuli: když máte jednotlivé User Stories rozpadnuté na tasky, stačí obvykle ukázat na jeden lísteček, který člen týmu dokončil, a na druhý, který dokončil dnes. Jednoduché a jasné. Ostatně zkušenější týmy zmíněné tři otázky přeskakují a Standup probíhá formou diskuse.

Psychologicky to buduje individuální závazek. Já jsem ráno slíbil týmu, že tuhle tasku dokončím, tedy musím udělat vše pro to, abych tým nezklamal. Co se stane, když se mi to nepovede? Nic kritického. Tým vnímá, že je tu něco, co jde o trochu pomaleji, než jsme si na začátku mysleli, a přemýšlí, jestli ještě stále věříme tomu, že cíle Sprintu dosáhneme. Zároveň taková situace iniciuje diskusi o tom, jestli by někdo neměl kolegoví pomoci s řešením či jestli by neměli vzhledem k týmovému závazku tuto situaci zohlednit a změnit priority.

Nezabíhejte do podrobností toho, co jste všechno dělali.

Klíčem k úspěšnému Standup meetingu je zaměřit se na to, co jste včera dokončili a co plánujete dnes dokončit vzhledem k dosažení cíle Sprintu.

Na co si dát se Scrum meetingem pozor:

- Věnujte se jen tomu, co jednotliví členové týmu dokončili, vyprávění o tom, co všechno museli udělat, aby se k výsledku dostali, obvykle nic nepřináší a Standup meeting zbytečně prodlužuje.
- Věnujte se jen tomu, co vede k cíli Sprintu. Ostatní aktivity se na Standupu nezmiňují.
- Jsou na meeting jednotliví členové připraveni? Vědí, co dokončili včera, a mají představu, na čem budou pracovat dnes a co dokončí?

Omezte detailní diskuse o konkrétním problému nebo jeho technickém řešení. Problém identifikujte a následnou diskusi odložte, ideálně hned po Scrum meetingu. A to i tehdy, když si myslíte, že je potřeba, aby se jí zúčastnili všichni členové týmu.

Když jednotliví členové stojí u Scrum tabule, je meeting výrazně rychlejší a efektivnější, než když sedí na svých místech u počítačů.

Scrum meeting není status meetingem pro Scrum Mastera, Product Ownera ani manažera.

Podstatou úspěšného Scrum meetingu je důraz na práci, kterou jednotliví členové dokončili. Posiluje se tím jak vnímání týmového závazku, tak i schopnost odhadovat, co který člen týmu dokončí.

Product Owner je vítán a je obvykle užitečný, aby se zúčastnil, ale je jen v roli pozorovatele. Zrovna tak ostatní návštěvníci.

Souvislosti:

(co budete ještě potřebovat zavést, detailní informace jsou v dalších kapitolách)

Sprint a Sprint Goal: Standup meeting nebude fungovat bez důrazu na společný cíl. Jednotliví členové týmu ho budou brát jen jako reporting Scrum Mastera

- Omezte detailní diskuse o konkrétním problému nebo jeho technickém řešení. Problém identifikujte a následnou diskusi odložte, ideálně hned po Scrum meetingu. A to i tehdy, když si myslíte, že je potřeba, aby se jí zúčastnili všichni členové týmu.
- Když jednotliví členové stojí u Scrum tabule, je meeting výrazně rychlejší a efektivnější, než když sedí na svých místech u počítačů.
- Scrum meeting není status meetingem pro Scrum Mastera, Product Ownera ani manažera.
- Podstatou úspěšného Scrum meetingu je důraz na práci, kterou jednotliví členové dokončili. Posiluje se tím jak vnímání týmového závazku, tak i schopnost odhadovat, co který člen týmu dokončí.
- Product Owner je vítán a je obvykle užitečné, aby se zúčastnil, ale je jen v roli pozorovatele. Zrovna tak ostatní návštěvníci.

Souvislosti:

(co budete ještě potřebovat zavést, detailní informace jsou v dalších kapitolách)

Sprint a Sprint Goal: Standup meeting nebude fungovat bez důrazu na společný cíl. Jednotliví členové týmu ho budou brát jen jako reporting Scrum Mastera

rovi / Product Ownerovi a bude jim v podstatě jedno, kdo na čem dělá.

User Stories: Standup meeting je o každodenním závazku jednotlivce. Naučte se rozpadat User Story na menší úlohy na cca jeden den. I práce na dokumentu jde nějak konkrétně pojmenovat. Zacílení na User Stories vám umožní sledovat průběh a monitorovat pravý stav projektu.

Tabule: Přehledná vizualizace na tabuli Standup meetingu vždy pomůže. I když už si nepamatujete, co kdo dělá, nebo jste meeting zmeškali, můžete se kdykoli podívat. Časem poznáte na první pohled, jak si tým stojí a jak je pravděpodobné, že svůj závazek dodrží.

Retrospektiva

ednou z velmi obecných agilních praktik, která je pří-
osná i v neagilních týmech a prostředích, je beze-
poru retrospektiva. Je to efektivní nástroj na získávání
ětné vazby a její formát můžete použít v libovolném
ontextu pro implementaci zlepšování a inovací, a to
kvírámcí týmů, tak i jednotlivců. V extrémním případě
se dalo říct, že když nevíte, jak s agilními metodami
čít, začněte retrospektivou.

rykněte si na pravidelnou zpětnou vazbu a naučte se
avidelně challengovat a měnit své procesy, praktiky
zyklosti. A s pomocí retrospektivy si teprve navrh-
ete, jak má váš agilní proces vypadat.

Retrospektiva má obvykle několik fází:

- Úvod**, kde si potvrdíme aktuální program.
- Sběr dat** je sekce, kde se snažíme posbírat co nejvíce informací o současném stavu: co fun-
guje a co by šlo naopak zlepšit.
- Hlubší porozumění informacím**, kde se sna-
žíme identifikovat příčiny a pochopit podstatu
identifikovaného problému či prostoru pro
zlepšení. Dobrá identifikace příčin je klíčem
úspěšné retrospektivy.
- Brainstorming možnosti** je základním staveb-
ním kamenem. Bez konkrétních akcí, kterými
můžeme danou oblast adresovat, se problémy

budou pořád opakovat a tým po čase ztratí
zájem o nich mluvit: Zbude jen pachut a nízká
motivace a slova „vždyt s tím stejně nic nenadě-
láme, tak proč se vůbec scházet“.

- Shrnutí konkrétních akcí** – závěr retrospektivy
je klíčový, shrneme v něm konkrétní kroky pro
změnu či zlepšení, ujednání, dohody. Bez kon-
krétních akcí na další Sprint odsouhlasených
celým týmem se retrospektiva stane jen poví-
dáním, bez možnosti cokoli ovlivnit, a ztratí tak
svou důležitost a smysl.

Jak začít?

Existuje mnoho formátů jak vést retrospektivu, ale nej-
častějším způsobem získání dat je takzvané kolečko.

Tým se posadí do místnosti kolem stolu. Facilitátor
meeting uvede a nechá každého člena týmu postupně
odpovědět na následující otázky:

- co se mi líbilo a v čem bych chtěl pokračovat
- co se mi nelíbilo a chtěl bych to změnit
- co bych zavedl nového

Retrospektiva je o pocitech, o tom, jak jsem se cítil,
co jsem si myslel. Není to žádná univerzální pravda.
Obvykle facilitátor meetingu pošle dokola tužku nebo
nějaký jiný objekt. Mluví jen ten, kdo má tužku. Když
domluví, předá dál. Nikdo ho nepřerušuje, nikdo s ním
nepolemizuje. Je to jen jeho pocit.

Sprint Planning

Když už tým rozumí Backlogu a zná priority pro následující Sprint, můžeme začít plánovat. Plánuje celý tým. Kvalitně provedený Backlog Refinement je nutnou podmínkou úspěchu, bez porozumění vizi produktu, prioritám, celkovému business kontextu a jednotlivým položkám backlogu tým plánovat nedokáže.

planning začínáme tím, že Product Owner představí sprint Goal. Ani ten by však neměl být překvapením, neboť o něm již jistě mluvili na Backlog Refinementu. Tým potom vybírá, které User Stories mu nejlépe pomůžou k Sprintu dosáhnout. Při výběru zvažuje nejen business hodnotu vzhledem k cíli Sprintu, ale i kapacitu. Vybrané položky se stávají Sprint Backlogem.

Scrum Master meeting moderuje tak, aby si tým uvědomil, kolik User Stories již plánuje dokončit a jestli je takový plán realistický, ale zároveň je motivující a pomáhá týmu ujasnit si Sprint Goal s Product Owne-rem. U začínajících týmů je obvykle Scrum Master méně aktivní a meeting moderuje tak, aby si tým uvědomil, k čemu všemu se opravdu zavazuje. Začneme tím, že Scrum Master se ptá se týmu, kterou User Story určitě dokončí. Tým vybere a dá na tabuli první. Scrum Master pokračuje a ptá se: Když dokončíte US1, kterou ještě další byste mohli dokončit? Tým vybírá další. Scrum Master znovu zrekapituluje závazek a ptá se: Když dokončíte US1 a US2, co ještě stihnete dokon-

čit? Po chvíli už tým není jednotný a dohaduje se, zda další User Story přidat, či ne; to je obvykle místo blízké finálnímu závazku.

V ten moment je dobré sečíst body vybraných User Stories a porovnat součet s rychlostí, které tým dosahoval za předchozí Sprints. Dobře fungující týmy mají rychlost stabilní. Zároveň je dobré se u začínajících týmů ptát, jestli mají všichni členové týmu pocit, že mají dost práce, a naopak jestli někdo nemá pocit, že je toho na něj v daném Sprintu moc.

Když tým dojde k závazku, necháme je ještě chvíli nad Sprint Backlogem diskutovat. V minimální variantě Product Owner a Scrum Master odejdou na 15 minut třeba na kafe a nechají tým, aby se v klidu nad Sprint Backlogem zamyslel. Tým během tohoto času začne rozkládat User Stories na menší úkoly. Když se Scrum Master a Product Owner vrátí, ptají se ještě jednou, jestli je takto sestavený Sprint Backlog finální, a případně akceptují jeho změnu. U delších Sprintů a složitějších produktů může tato doba být až jeden den.

Ze začátku vám může planning trvat déle, ale během prvních pár Sprintů by se měl ustálit na rozumném čase. U dvoutýdenního Sprintu, kde tým zná Product Backlog a připravuje se průběžně plánování, netrvá déle než 30 minut.

Součástí planningu je i to, že tým vybrané User Stories rozloží na úkoly o velikosti cca 1 den. Některé týmy tuto aktivitu dělají během prvního dne Sprintu, ale obecně platí, že čím dříve, tím lépe. Jinak ztrácíte přehled, co je v jakém stavu a co nám ještě chybí k dokončení jednotlivých User Stories. Ohodnocovat jakkoli úkoly je zbytečné. Udělejte je na cca 1 den a z tabule pak vidíte, jak na tom jste.

Úkoly, které vám trvaly déle, můžete označit třeba tečkou za každý další den, ale neztrácejte čas jejich dopředným odhadováním ani v hodinách, ani v bodech. Best Practice říká, že by úkol neměl být větší než na dva dny. Když zjistíte že se vám práce vleče více, než jste čekali, rozdělte ho. Zároveň si klidně poznačte jako úkol malé drobnosti, na které nechcete zapomenout.

Sprint Goal je během Sprintu neměnný. Product Owner by měl tedy dobře zvážit priority dopředu. Jedinou možnost, kterou má v případě, že se mu cíl výrazně změní, je Sprint zastavit a naplánovat nový Sprint znovu. V tom případě se všechna práce zastavuje, User Stories se vrací do Product Backlogu a tým plánuje nový Sprint. Asi nemusíme příliš upozorňovat, že je to hodně drahá možnost. Sprint by měl být tak krátký, abyste vždy mohli počkat na další Sprint a tam novou prioritní funkcionalitu zařadit. Když zapojíte zákazníka do Refinementu a budete s ním dobře komunikovat, půjde to snadno.

Co dělat když:

Planning je moc dlouhý: Tým není pravděpodobně na meeting připravený a není schopen naplánovat Sprint. Vedou se nekonečné diskuse o User Stories. Zlepšete nebo začněte dělat Backlog Refinement. Zvažte zavedení pravidelného Backlog Groomingu, kde budete několikrát za Sprint (1 h týdně) spolu s týmem procházet vršek Backlogu, aby mu obě strany rozuměly. Výsledkem může být jiné rozdělení User Stories nebo jen diskuse s týmem a pochopení kontextu.

Tým plánuje moc: Tým má často velké oči a plánuje na ideální Sprint, který ovšem nikdy nenastane. „Kdyby nás nic nevyrušovalo a měli bychom úplně všechny informace, tak bychom to stihli.“ Jeden nebo dva sprinty to není problém. Výsledek rozeberte s týmem na retrospektivě a dohodněte se, co dál. I neobvyklých a výjimečných věcí se děje každý Sprint přibližně stejně. Změřte si reálnou rychlost týmu a plánujte další Sprint na podobnou hodnotu.

Tým nedokončí všechny položky Sprint Backlogu: Nedokončené User Stories bez ohledu na jejich případnou rozpracovanost jdou zpět do Product Backlogu. Je jen na Product Ownerovi, zda je zařadí do dalšího Sprintu nebo ne. Tým by se tedy měl snažit User Stories dokon-

čovat, jak jen nejrychleji to jde, a nemít příliš rozpracovaných User Stories v průběhu Sprintu. Zároveň by tým neměl začínat User Story, kterou už nestihne dokončit, a raději se věnovat zlepšením či Refactoringu.

Souvislosti:

(co budete ještě potřebovat zavést)

Backlog Refinement: Dobré porozumění Backlogu je klíčem úspěchu Sprint Planningu.

Sprint: Tým plánuje práci vždy na další Sprint, během Sprintu se pak Sprint Goal nemění.

Sprint Backlog: Výstupem Sprint Planningu je Sprint Goal, Sprint Backlog a dohoda o tom, jak budeme spolupracovat.

Tabule: Součástí planningu je i rozpad jednotlivých User Stories na úkoly. Dejte je rovnou na tabuli, ať všichni vidí, co ještě chybí k dokončení User Story.

Product Owner: Představuje Sprint Goal a pomáhá týmu vybrat konkrétní User Stories do Sprint Backlogu.

čovat, jak jen nejrychleji to jde, a nemít příliš rozpracovaných User Stories v průběhu Sprintu. Zároveň by tým neměl začínat User Story, kterou už nestihne dokončit, a raději se věnovat zlepšením či Refactoringu.

Souvislosti:

(co budete ještě potřebovat zavést)

Backlog Refinement: Dobré porozumění Backlogu je klíčem úspěchu Sprint Planningu.

Sprint: Tým plánuje práci vždy na další Sprint, během Sprintu se pak Sprint Goal nemění.

Sprint Backlog: Výstupem Sprint Planningu je Sprint Goal, Sprint Backlog a dohoda o tom, jak budeme spolupracovat.

Table: Součástí planningu je i rozpad jednotlivých User Stories na úkoly. Dejte je rovnou na tabuli, ať všichni vidí, co ještě chybí k dokončení User Story.

Product Owner: Představuje Sprint Goal a pomáhá týmu vybrat konkrétní User Stories do Sprint Backlogu.

Sprint Review

rovna tak jako na začátku Sprintu se ptáme zákazníka, o by chtěl vidět, na konci každého Sprintu nadchází chvíle, kdy zákazníkovi ukazujeme, co jsme dokončili. během Sprint Review se snažíme získat zpětnou vazbu a to, jak jsme adresovali cíl Sprintu. Obecně zákazník nezájímají jednotlivé User Stories. Ukažte mu celý inkrement produktu. Čím více zapojíte zákazníka a dáte mu produkt vyzkoušet, tím více bude i v budoucnu součástí vašeho týmu a bude mít produkt rád. Dopřejte mu zážitek z funkcionality, kterou jste pro něj napsali.

Zákazníkem je v tomto kontextu kdokoliv, kdo má a projektu nějaký zájem. Nemusí to tedy vždy být oncový uživatel. Na některých produktech budete mít pouze interní zákazníky, někde bude možné pozvat přímo uživatele.

ozvěte na Sprint Review zástupce zákazníků a uživatelů, at' dostanete kvalitní zpětnou vazbu.

pučástí ukazované hodnoty jsou pouze dokončené User Stories. Tedy ty, které fungují, jsou otestovány, zredovány, popsány v dokumentaci a akceptovány Product Ownerem. Nestačí tedy, když je jen nakódovaná vývojář říká, že funguje. Často se proto používá výraz „done-done“. Nestačí, když je funkcionality hotová, musí být „zcela hotová“. Jestli je některá User Story moc

velká, rozdělte ji ještě v průběhu Sprintu na menší User Stories. Skoro hotová User Story na Sprint Review nepatří. Nespĺňuje Definition of Done, a mohla by tak vzбудit mylná očekávání, že vše funguje, i když to vlastně funguje jen napůl. A to určitě nechcete.

Nechte prezentovat přímo tým. Chce to možná trochu cviku, ale zvyšuje to motivaci jednotlivých členů týmu zrovna tak jako jejich pochopení zákazníka a jeho požadavků. Tým také obvykle nejlépe ví, jak systém funguje a co ukázat. Product Owner je v roli posluchače a pozorovatele a dává pozor, aby mu žádný feedback neutekl.

Nezapomeňte že cílem Sprint Review je zpětná vazba, nikoli status ani akceptace. Product Owner dokončené User Stories akceptuje již v průběhu Sprintu, aby zákazníkovi šla již již funkcionality, která je za Scrum tým v pořádku, a nemarnili jsme jeho čas na věci, které dokážeme odchytit sami.

Sprint review se zaměřuje na cíl Sprintu, není to technické review ani nemusí detailně ukazovat jednotlivé položky Backlogu.

Cílem Sprint Review je získat zpětnou vazbu na dodanou business hodnotu.

Souvislosti:

(co budete ještě potřebovat zavést)

Sprint: Sprint dává Sprint Review pravidelnost.

Zákazník: Prezentujete funkcionalitu zákazníkovi – ať už internímu, nebo externímu. Zákazníkem v tomto kontextu je kdokoliv, kdo má na výsledku nějaký zájem.

Definition of Done: Domluvte se předem, jak přesně bude vypadat Definition of Done, tedy v jaké finální kvalitě jednotlivé User Stories prezentujete.

Tým: Tým je zodpovědný za dodání Sprint Backlogu. Selže-li jednotlivec, selhal i celý tým.

Agilní program

Scrum je proces. Budete dobré spolupráce v týmu, širší flexibilitu, efektivitu práce alespoň některých prvků je svým způsobem bez z metody Extreme Programming, by měly být omezeny.

Pair Programming

Jednoduchá myšlenka. Když máte méně chyb, je zábrana sám a překvapivě je to v praxi? Dva vývojáři sedí u jedné klávesnici a jeden myšlí, kontroluje, přičítá to opravdu rychlejší než to samostatně. Navíc mají nižší procento chyb.

Přestože Scrum zavádí nezávislé týmy, často jsou týmy jen dočasně, když se týmu nebo když řeší větší problémy. Právní princip Pair Programming součástí Scrum procesu. Je velmi nízkou znalostí.