

# Základní modely životního cyklu software, softwarový proces, metodika.

Wednesday, May 29, 2013 4:49 PM

**Proces** – systematická série akcí vedoucí k určitému výsledku

**Životní cyklus** (meta proces?) – proces od zahájení vývoje až po vyřazení z provozu

## Metodika

definovaný proces pro konkrétní účel, tj. fáze, aktivity, role, artefakty, milníky atd. jsou dobře popsány (z přednášky)

- souhrn doporučených praktik a postupů, pokrývajících celý životní cyklus vytvářené aplikace

Z <http://cs.wikipedia.org/wiki/Metodika> >

- o Booch method
- o SSADM
- o Rational Unified Process
- o SCRUM
- o ...

- UML není metodika!

## Softwarový proces

systematická série fází a **aktivit**, souvisejících **rolí** a **artefaktů** vedoucí k vyšší pravděpodobnosti úspěšného vytvoření potřebného software.

- Výsledek = kvalitní software
- Členění: fáze, **aktivita**
  - o *Technické aktivity: komunikace, plánování, modelování, konstrukce, nasazení*
  - o *Podpůrné aktivity: řízení, kontrola kvality, správa konfigurace, dokumentace*
- Mezivýsledky: **artefakty**
  - o *Technické: specifikace, dokumentace, testy, modely, ...*
  - o *Komunikační: specifikace, plán*
  - o *Obchodní: plán, rozpočet, produkt*
- Činitel: **role**
  - o *Technické role: analytik, architekt, vývojář, tester, databázista, ...*
  - o *Manažerské role: team leader, šéf vývoje, šéf projektů, CEO, CIO, ...*
  - o *Podpůrné role: poradce, lektor, uživ. podpora, dokumentace, ...*

Varianty SW procesu (modely životního cyklu software):

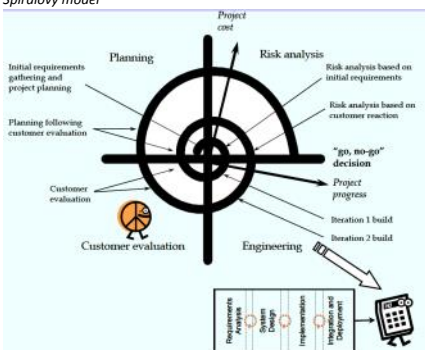
### Varianty a příklady procesů:

Společná snaha = snížení rizika chaotického postupu

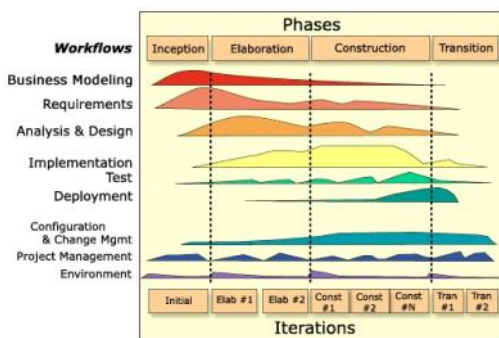
- **Řízené plánem** (sledovat plán – hra na jistotu)
  - o Vodopád, V-model
  - o Kontext neměnný, zadání a technologie zřejmé, rozsah malý
- **Řízené riziky** (omezovat rizika)
  - o Průzkumník/prototypování, Spirálový model
  - o Kontext zřejmý, zadání a/nebo technologie nejasné
- **Řízené změnou** (adaptace na změnu)
  - o Iterativní – RUP, agilní – SCRUM
  - o Kontext proměnlivý, zadání a/nebo technologie nejasné

## Základní modely životního cyklu

- **Sekvenční** - *velký třesk* (vztažené na celý produkt, naplánované pro celý projekt, oddělené meziprodukty)
  - o *Vodopádový model*
- **Cyklický** - *opakování technických aktivit, přírůstky* (roste znalost, funkcionalita, kvalita, ...)
  - o *Spirálový model*



- o RUP



- o *Inkrementální (přírůstek je nějaká část (funkcionalita) produktu, produkt je funkční až na konci vývoje)*
- o *Iterativní (na konci každé iterace je funkční produkt, např. Na počátku jen se základní funkcionalitou, ale v dalších iteracích se funkcionalita postupně rozšiřuje)*
- **Agilní - evoluce**
  - o extrémní programování XP

## Základní modely životního cyklu software, softwarový proces, metodika.

### Metodika

Konkrétní proces, má stanovené fáze, milníky, artefakty...

### Základní modely životního cyklu software

Životní cyklus = proces od zahájení vývoje až po vyřazení z provozu

Sekvenční -> velký třesk

- Waterfall = vodopádový model, fáze jdou postupně od inception, elaboration, implementation, transition. Na konci je tzv. velký třesk, předání zákazníkovi – může dopadnout špatně, výsledek se nemusí shodovat s tím, co zákazník chtěl

Cyklický -> přírůstky/evoluce

- Spirálový model

Agilní -> Evoluce

Iterativní a inkrementální

- Pracuje se postupně v iteracích, na konci každé iterace je funkční produkt, jednotlivé iterace jsou jako samostatné malé projekty, které sestávají z několika fází

RUP – rational unified proces – od IBM, je to Framework, složitá implementace, častou chybou je, že se společnost rozhodne implementovat ten Framework celý – to je ale špatně, společnost by si měla vybrat jen to, co se jí hodí

OpenUP – částečně postaven na RUP, bere si z něj ale jen to nejnútnejší pro vývoj software, je open source, distribuce formou wiki

SCRUM – daily standups, sprinty (30 dní), iterativní a inkrementální, plánuje jen následující iteraci, není předem jisté kdy projekt skončí

### Softwarový proces

Softwarový proces je systematická série aktivit, artefaktů a souvisejících rolí vedoucí k vyšší pravděpodobnosti úspěšného dokončení výroby software.

Výsledek = kvalitní software

Aktivity (Technické – modelování, programování, nasazení..., Komunikační – analýza, specifikace, ...)

Artefakty (Dokument specifikace, Test protocol...)

Role (Technické = analytik, team leader, developer, architekt..., Manažerské = CEO, CIO..., Podpůrné = poradce, lektor...)

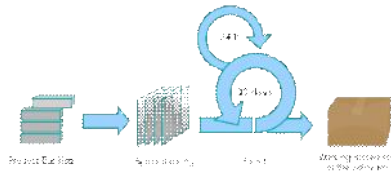
**Řízené plánem** – RUP

**Řízené riziky** – prototypování

**Řízené změnou** – SCRUM

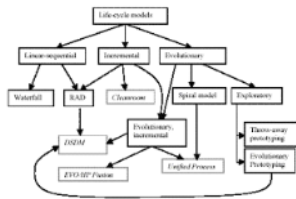


o SCRUM



o Důraz na manifest:

- Jednotlivci a interakce před procesy a nástroji
- Fungující software před vyčerpávající dokumentací
- Spolupráce se zákazníkem před vyjednáváním o smlouvě
- Reagování na změny před dodržováním plánu



## Agile vs Iterative vs Spiral

Agile Methods are a family of techniques that includes iterative development. Spiral is a type of iterative method that extends Waterfall methods. Spiral tends to be associated with longer iterations and normally involves a full development life cycle including extensive analysis and documentation.

The modern concept of Agile developed from the Agile Manifesto. Various practitioners from different software disciplines came together to articulate a common set of principles. These ideas incorporated concepts developed from extreme programming, unified process, scrum, test driven development, iterative programming, pair programming, continuous build, and others. To some degree, all of these so-called "practices" assist in the implementation of an Agile method. Every Agile project is somewhat different based on the composition of the team and the familiarity with the various practices used to build software.

Iterative development is a concept that can be applied across the continuum of methodologies ranging from extreme Agile to extreme Waterfall. Iterative development is characterized by the completion of a working system within a subset of time for the whole project and the repeated refinement of the working system based on cycles of development. In extreme programming iterations tend to be short, i.e. one to three weeks. Scrum based development methods advocate longer cycles of 2 to 6 weeks. Spiral development methods may have much longer iterations lasting several months.

Whether or not an iteration can be categorized as "agile" or "spiral" depends on the components of the iteration. The extreme agile iteration has little or no documentation, has many tests, and has a quick cycle. It may focus on one area of the overall system. A spiral iteration generally incorporates a complete development life cycle including analysis, planning, implementation, testing, and delivery.

From <<http://agilemanagementresearch.blogspot.cz/2009/10/agile-vs-iterative-vs-spiral.html>>

From <<http://testconsultant.blogspot.cz/2008/08/development-models-and-impact-on.html>>

From <<https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>> #

# SCRUM

Thursday, June 6, 2013 11:26 AM

# The Scrum Guide

---

Průvodce Scrumem:  
Pravidla hry



*Jeff Sutherland*



*Ken Schwaber*

*říjen 2011*

*Vyvinuli a udržují Ken Schwaber a Jeff Sutherland  
Český překlad vytvořila agilia.cz*



Improving the Profession of Software Development



Cíl průvodce Scrumem .....	3
Scrum v kostce .....	3
Základní rámec .....	3
Příklad .....	3
Teorie Scrumu .....	3
Scrum Tým .....	4
Vlastník produktu .....	4
Vývojový tým .....	5
Scrum master .....	6
Činnosti ve Scrumu .....	7
Sprint .....	7
Plánovací schůzka .....	8
Daily Scrum .....	9
Vyhodnocení sprintu (Sprint Review) .....	10
Retrospektiva sprintu .....	10
Artefakty .....	11
Produktový backlog .....	11
Sprint Backlog .....	13
Přírůstek .....	13
Definice toho co je „hotovo“ .....	13
Závěr .....	14
Poděkování .....	14
Lidé .....	14
Historie .....	14





## Cíl průvodce Scrumem

Scrum, který je popsán v tomto průvodci, je rámec pro vývoj a údržbu složitých produktů. Tento průvodce popisuje role, činnosti a artefakty Scrumu a pravidla, která je drží dohromady. Autory Scrumu jsou Ken Schwaber a Jeff Sutherland, kteří také napsali anglickou verzi průvodce.

## Scrum v kostce

Scrum je nástroj, s jehož pomocí mohou lidé zvládnout složité adaptivní problémy a při tom dodávat produkty s vysokou přidanou hodnotou. Scrum je:

- jednoduchý
- srozumitelný
- extrémně obtížný pro dokonalé zvládnutí

Scrum je procesní rámec, který se používá k řízení vývoje složitých produktů od začátku devadesátých let. Scrum není proces pro samotný vývoj produktů, je to spíše procesní rámec, uvnitř kterého lze používat jiné procesy a techniky. Scrum zviditelňuje účinnost metod vašeho produktového řízení a vývoje, takže je možné je dále zdokonalovat.

## Základní rámec

Scrum se skládá ze Scrum týmů a přidružených rolí, činností, artefaktů a pravidel. Všechny součásti slouží určitému účelu a jsou nezbytné k tomu, aby bylo nasazení Scrumu úspěšné.

Konkrétní strategie nasazování Scrumu se liší případ od případu a jsou popsány jinde.

Vztahy a interakce mezi činnostmi, rolemi a artefakty určují pravidla Scrumu. Tato pravidla jsou popsána dále v dokumentu.

## Překlad

Průvodce byl přeložen z původní anglické verze, kterou poskytli Ken Schwaber a Jeff Sutherland. Na překladu se podíleli Robert Batůšek, Zdeněk Měrka, Martin Paviček a Michal Vallo. Jazykovou korekturu udělali Petr Smejkal a Petr Sobotka. Práce koordinoval Robert Batůšek.

## Teorie Scrumu

Scrum je založen na teorii řízení empirických procesů neboli empirismu. Empirismus tvrdí, že znalost pochází ze zkušenosti a rozhodování založeném na tom, co je známo. Scrum využívá iterační a inkrementální přístup k optimalizaci předvídatelnosti a řízení rizik. Každá implementace řízení empirického procesu stojí na třech pilířích: transparentnosti, kontrole a adaptaci.

### Transparentnost

Důležité aspekty procesu musí být viditelné těm, kteří mají vliv na výsledek. Transparentnost vyžaduje, aby tyto aspekty používaly společný standard, takže pozorovatelé budou rozumět tomu, co vidí.

Například:

- Všichni účastníci musí používat společný jazyk, kterým popisují proces.







- Ti, kteří se účastní prací, musí používat společnou definici toho, co je „hotovo“.

Transparentnost zajišťuje, aby všechny aspekty procesu, které mají vliv na výsledek, byly stále viditelné a na první pohled srozumitelné pro lidi, kteří mají výsledky dodat. To znamená: jestliže ten, kdo proces kontroluje, považuje něco za hotové, musí se to plně shodovat s tím, jak je „hotovo“ definované.

### Kontrola

Uživatelé Scrumu musí kontrolovat artefakty a postup směrem k cíli tak často, aby se daly odhalit nepřijatelné odchylky v procesu. Frekvence kontroly nicméně nesmí být tak vysoká, aby stála v cestě skutečné práci. Kontrola je nejužitečnější, když ji provádějí kvalifikovaní lidé přímo na místě, kde se pracuje.

### Adaptace

Jestliže revizor na základě kontroly rozhodne, že jeden nebo více aspektů procesu jsou mimo přijatelné hranice a že výsledný produkt bude nepřijatelný (neakceptovatelný), pak je nutné tento proces adaptovat. Změna musí být provedena co nejdříve, aby byla minimalizována budoucí odchylka.

Ve Scrumu jsou definovány čtyři formální body pro kontrolu a adaptaci (viz také kapitola Činnosti ve Scrumu):

- Plánování sprintu
- Daily Scrum
- Vyhodnocení sprintu
- Retrospektiva sprintu

### Obsah Scrumu

Scrum je rámec navržený pro podporu vývoje složitých produktů. Scrum se skládá z Scrum týmů a přidružených rolí, činností, artefaktů a pravidel. Každá součást rámce slouží určitému účelu a je nezbytná k tomu, aby bylo nasazení Scrumu úspěšné.

## Scrum Tým

Scrum tým se skládá z vlastníka produktu, vývojového týmu a Scrum mastera. Scrum týmy jsou sebeorganizující a multifunkční. Sebeorganizující týmy si samy volí, jak provedou práci, nejsou tedy přímo vedeny nikým zvenčí. Multifunkční týmy mají všechny schopnosti potřebné k tomu, aby dokončily svou práci bez toho, že by musely čekat na někoho, kdo není součástí týmu.

Týmy doručují produkty iterativně a inkrementálně, a tím zvyšují šanci na to, že se jim dostane zpětné vazby. Postupné doručování v kvalitě „hotovo“ zajišťuje, že použitelná verze produktu je vždy k dispozici.

### Vlastník produktu

Vlastník produktu je zodpovědný za maximalizaci hodnoty produktu a práce vývojového týmu. Organizace, týmy i jednotlivci mohou volit různé cesty, jak toho dosáhnout.





Vlastník produktu je jediný člověk, který je zodpovědný za správu produktového backlogu, která zahrnuje tyto činnosti:

- Jasná formulace položek produktového backlogu.
- Uspořádání položek produktového backlogu tak, aby co nejlépe odrážel produktovou vizi a cíle.
- Zajišťování toho, aby vývojový tým dodával vždy nejlepší hodnotu.
- Zajištění transparentnosti a dostupnosti produktového backlogu tak, aby bylo každému srozumitelné, co ukazuje a na čem bude Scrum tým v nejbližší době pracovat.
- Zajištění toho, že produktový backlog je transparentní, viditelný, jasný každému a ukazuje, na čem bude Scrum tým pracovat v nejbližší době.
- Zajištění toho, že vývojový tým dostatečně rozumí položkám v produktovém backlogu.

Výše popsanou práci může vykonávat vlastník produktu sám nebo může pověřit vývojový tým, aby ji vykonával on. Vlastník produktu však v každém případě zůstává zodpovědnou osobou.

Vlastník produktu je osoba, nikoli komise. Vlastník produktu může reprezentovat zájmy komise, ale pokud někdo chce změnit pořadí položek v produktovém backlogu, musí o tom přesvědčit vlastníka produktu.

Aby mohl být vlastník produktu úspěšný, celá organizace musí respektovat jeho nebo její rozhodnutí. Rozhodnutí vlastníka produktu jsou viditelná v obsahu a pořadí položek produktového backlogu. Nikdo jiný není oprávněn pověřit vývojový tým, aby pracoval podle jiných požadavků, a vývojový tým nemá dovoleno pracovat podle toho, co řekne někdo jiný.

## Vývojový tým

Vývojový tým se skládá z profesionálů, kteří dodávají přírůstek „Hotového“ produktu na konci každého sprintu. Přírůstek produktu vytvářejí pouze členové vývojového týmu.

Organizace vytváří vývojové týmy a pověřuje je, aby spravovali svou vlastní práci. Vzniklý synergický efekt vede ke zvýšené efektivitě práce vývojového týmu. Vývojové týmy mají následující charakteristiky:

- Jsou sebeorganizující. Nikdo (dokonce ani Scrum master) neříká vývojovému týmu, jak má přeměnit produktový backlog v přírůstek produktu.
- Vývojové týmy jsou multifunkční, mají všechny schopnosti potřebné k vytvoření přírůstku produktu.
- Ve Scrumu nemají členové vývojového týmu jiný titul než vývojář bez ohledu na to, jakou práci vykonávají. Z tohoto pravidla nejsou žádné výjimky.
- Jednotliví členové vývojového týmu mohou mít svou specializaci, ale zodpovědnost za práci má tým jako celek.
- Vývojové týmy neobsahují podtýmy, které se věnují konkrétním oblastem, jako např. testování nebo analýza.





### Velikost vývojového týmu

Ideální vývojový tým je dost malý na to, aby zůstal flexibilní a dost velký na to, aby byl schopen dokončit smysluplnou práci. Týmy s méně než třemi členy nemohou naplno využít interakcí mezi členy, což vede pouze k malému zlepšení produktivity. Menším vývojovým týmům se také může stát, že nebudou mít všechny kompetence potřebné k dokončení sprintu a doručení přírůstku produktu. Týmy s více než devíti členy vyžadují příliš mnoho koordinace. Takové týmy jsou příliš složité na to, aby bylo možné použít k jejich řízení empirický proces.

Vlastník produktu a Scrum master se do těchto počtů nepočítají (pokud ovšem nepracují na úkolech z produktového backlogu).

### Scrum master

Scrum master je zodpovědný za osvojování a dodržování pravidel Scrumu. Scrum masteři trvají na tom, aby týmy dodržovaly jak ducha teorie Scrumu, tak i jeho techniky a pravidla. Scrum master zaujímá roli vedoucího týmu, ovšem je to vedoucí, který týmu hlavně slouží.

Scrum master pomáhá nečlenům vývojového týmu rozpoznat, které jejich interakce s týmem jsou prospěšné a které ne. Pomáhá také každému změnit své chování tak, aby bylo pro vývojový tým maximálně prospěšné.

### Služby Scrum mastera vlastníkovu produktu

Scrum master pomáhá vlastníkovu produktu těmito způsoby:

- Hledá techniky pro efektivní správu produktového backlogu.
- Jasně komunikuje produktovou vizi, cíle a položky produktového backlogu vývojovému týmu.
- Učí vývojový tým, jak vytvořit jasný a stručný produktový backlog.
- Rozumí tomu, jak vytvářet dlouhodobou produktovou vizi v empirickém prostředí.
- Rozumí tomu, jak aplikovat principy agilního vývoje.
- Moderuje podle potřeby všechny schůzky ve Scrumu.

### Služby Scrum mastera vývojovému týmu

Scrum master pomáhá vývojovému týmu těmito způsoby:

- Vede vývojový tým směrem k sebeorganizovanosti a multifunkčnosti.
- Učí a vede vývojový tým k tomu, aby vytvářel produkty s vysokou přidanou hodnotou.
- Odstraňuje překážky, které brání vývojovému týmu v práci.
- Moderuje podle potřeby všechny schůzky ve Scrumu.
- Koučuje vývojový tým v prostředí organizací, ve kterých ještě Scrum není správně pochopen a přijat.

### Služby Scrum mastera organizaci

Scrum master pomáhá organizaci několika způsoby:

- Školí organizaci v osvojování Scrumu.
- Plánuje implementaci Scrumu v organizaci.





- Pomáhá zaměstnancům a ostatním zúčastněným stranám pochopit a osvojit si Scrum a empirický vývoj produktu.
- Iniciuje změny, které vedou k vyšší produktivitě produktového týmu.
- Spolupracuje s ostatními Scrum mastery na zavádění Scrumu v organizaci.

## Činnosti ve Scrumu

Předepsané činnosti Scrumu zajišťují pravidelnost a minimalizují potřebu dalších, Scrumem nedefinovaných schůzek. Scrum používá časově ohraničené (time-boxed) činnosti, tzn. že každá činnost má určenou svou maximální délku trvání. Plánování tak zabere přiměřenou dobu a v plánovacím procesu nedochází ke zbytečnému plýtvání časem.

Scrum sám o sobě je jen souhrnem předepsaných činností. Každá činnost je přitom i příležitostí ke kontrole a adaptaci některého aspektu Scrumu. Tyto činnosti jsou navrženy tak, aby umožnily důležitou transparentnost a kontrolu. Vynechání kterékoliv z těchto činností má za následek snížení transparentnosti a ztrátu kontroly a možnosti adaptace.

## Sprint

Podstatou Scrumu je Sprint v délce trvání jednoho měsíce, popřípadě kratší. Během Sprintu je vytvořen (v kvalitě „hotovo“) potenciálně nasaditelný přírůstek produktu. Sprints mají v rámci celého vývoje produktu shodnou délku. Nový Sprint vždy začíná ihned po dokončení předchozího Sprintu.

Sprints se skládají z plánovací schůzky (Sprint Planning Meeting), denních schůzek (Daily Scrums), vlastních vývojových prací, vyhodnocení sprintu (Sprint Review), a retrospektivy (Sprint Retrospective).

Během Sprintu:

- Se neprovádí žádné změny ovlivňující cíl sprintu (Sprint Goal).
- Se nemění složení vývojového týmu.
- Se nesnižuje kvalita cíle Sprintu.
- Může být mezi vlastníkem produktu a vývojovým týmem znovu projednán a upřesněn rozsah, jak bude popsáno v dalším textu.

Každý sprint můžeme považovat za projekt v (maximálně) měsíčním rozsahu. Stejně jako projekty, tak i sprints používáme k „vytvoření něčeho“. Součástí každého Sprintu je popis toho, co má být vytvořeno (tj. návrh cílového produktu a flexibilní plán, který bude sloužit jako vodítka pro provádění prací), dále samotná práce a výsledný produkt.

Sprints jsou limitované jedním kalendářním měsícem. Trval-li by sprint příliš dlouho, mohla by se mezitím změnit definice cílového produktu, zvýšit se jeho složitost a mohla by vzrůstat rizika. Sprints přinášejí předvídatelnost – přinejmenším jednou měsíčně zajistí provedení kontroly a adaptace procesu. Sprints také omezují riziko nekontrolovaného vzrůstu nákladů na maximálně jeden kalendářní měsíc.







### Zrušení Sprintu

Sprint může být před uplynutím plánovaného času zrušen. Jen vlastník produktu má pravomoc zrušit sprint, ovšem může to udělat na popud vývojového týmu, Scrum mastera či ostatních zúčastněných stran.

Sprint může být zrušený v případě, kdy cíl Sprintu zastará. Toto může nastat např. v případě změny vnitrofiremní strategie, změny situace na trhu nebo změn technologických podmínek. Obecně by Sprint měl být zrušený v případech, kdy za daných okolností již nedává smysl. Díky krátkým Sprintům je však nutné zrušit Sprint jen výjimečně.

Je-li sprint zrušený, jsou všechny „hotové“ položky produktového backlogu vyhodnoceny. Jsou-li hotové položky potenciálně nasaditelné, tak je vlastník produktu většinou akceptuje. Veškeré nehotové položky produktového backlogu znovu projdou procesem odhadování náročnosti a vrátí se zpět do produktového backlogu. Práce odvedená na těchto položkách rychle zastarává a mnohdy musí znovu projít ohodnocením náročnosti.

Zrušení sprintu spotřebuje určité zdroje, protože všichni musí na plánovací schůzce dalšího sprintu přeplánovat. Ke zrušení sprintu dochází jen výjimečně, a často je to pro Scrum tým traumatizující.

### Plánovací schůzka

Práce, která má být vykonána během sprintu, je plánována na plánovací schůzce (Sprint Planning Meeting). Tento plán je vytvářen v součinnosti celého Scrum týmu.

Plánovací schůzka je časově ohraničená – pro jednoměsíční sprint by měla zabrat nejvýše osm hodin. U kratších sprintů by tato činnost měla zabrat úměrně kratší dobu. Například dvoutýdenní sprint bude mít čtyřhodinovou plánovací schůzku.

Plánovací schůzka má dvě části; přičemž každá část je časově ohraničená: má trvat polovinu doby vyhrazené na plánovací schůzku. Plánovací schůzka odpovídá na dvě otázky - každá z částí na tu svou:

- Jaký přírůstek bude dodán na konci příštího sprintu?
- Jakou práci bude nutno vykonat pro vytvoření přírůstku?

### První část: co bude vytvořeno během sprintu?

V této části vývojový tým odhaduje, které všechny funkčnosti budou během sprintu vyvinuty. Vlastník produktu představuje vývojovému týmu seřazené položky produktového backlogu. Přitom se celý Scrum tým snaží společně porozumět obsahu sprintu.

Vstupními informacemi této schůzky jsou: produktový backlog, poslední přírůstek produktu, plánovaná kapacita vývojového týmu pro příští sprint a výkon vývojového týmu v předchozím sprintu. Počet položek produktového backlogu zařazených do sprintu závisí výhradně na rozhodnutí vývojového týmu. Jen vývojový tým může odhadnout, co je schopen během sprintu dokončit.

Poté, co vývojový tým odhadne, které položky produktového backlogu během sprintu dodá, tak Scrum tým definuje cíl sprintu (Sprint Goal). Cílem sprintu je něco, čeho má být (během sprintu) dosaženo realizací produktového backlogu, a vývojovému týmu dává odpověď na otázku „proč vlastně tento přírůstek realizujeme?“.





## Druhá část: jakým způsobem bude práce provedena?

Stanovil-li vývojový tým obsah sprintu, může začít uvažovat o tom, jak během sprintu vytvoří plánovaný přírůstek (jak jej převede do stavu "hotovo"). Položky produktového backlogu zařazené do tohoto sprintu spolu s plánem na doručování položek tvoří Sprint backlog.

Vývojový tým obvykle začíná návrhem systému a prací potřebných k přeměně produktového backlogu do fungujícího inkrementu produktu. Charakter prací se může během sprintu změnit, může se měnit i odhad pracnosti. Nicméně během plánovací schůzky by měl být naplánován dostatečný čas, který dle odhadu vývojového týmu bude potřebné odpracovat v nadcházejícím sprintu. Na konci plánovací schůzky jsou práce naplánované vývojovým týmem na první dny sprintu rozloženy na části o jednodenní (nebo menší) pracnosti. Aby mohl tým převzít zodpovědnost za provedení prací zařazených do sprint backlogu, tak se sebe-organizuje jak během plánovací schůzky, tak během samotného sprintu.

Vlastník produktu se může účastnit druhé části plánovací schůzky - pak může objasnit vybrané položky produktového backlogu a pomoci s dosažením jednotného pohledu na věc. Rozhodne-li vývojový tým, že má příliš mnoho nebo příliš málo práce, může s vlastníkem produktu znovu projednat položky sprint backlogu. Vývojový tým může přizvat k účasti i další lidi kvůli konzultacím týkajícím se tématu produktu nebo použité technologie.

Na konci plánovací schůzky by měl být vývojový tým schopen vlastníkovi produktu a Scrum masterovi vysvětlit, jakým způsobem hodlá jako sebe-organizující tým dosáhnout cíle sprintu a vytvořit očekávaný přírůstek.

## Cíl Sprintu

Definujeme-li cíl Sprintu (Sprint Goal), vývojový tým má vymezené určité hranice (týkající se funkčnosti implementované během sprintu), v jejichž rámci se může pohybovat.

Tento cíl má vývojový tým během své práce na paměti. Aby splnil cíl, implementuje funkcionalitu a technologii. Pokud se ukáže, že pracnost bude jiná, než tým předpokládal, pak jedná s vlastníkem produktu o rozsahu Sprint backlogu.

Cílem sprintu může být jeden z milníků vývoje produktu.

## Daily Scrum

Daily Scrum je 15-ti minutová schůzka vývojového týmu určená pro synchronizaci aktivit a vytvoření plánu na dalších 24 hodin. Na této schůzce je kontrolována práce vykonaná od posledního Daily Scrumu a je vytvářen odhad práce, která by mohla být vykonaná do dalšího Daily Scrumu.

Pro zjednodušení organizace se Daily Scrum koná každý den ve stejný čas a na stejném místě. Na schůzce každý člen vývojového týmu odpovídá na otázky:

- Co bylo dokončeno od doby poslední schůzky?
- Co bude dokončeno do termínu další schůzky?
- Jaké překážky nám stojí v cestě?

Vývojový tým pomocí Daily Scrumu vyhodnocuje svůj postup směrem k cíli sprintu. Vyhodnocuje, zda dosavadní postup směřuje k dokončení sprint backlogu. Daily Scrum zvyšuje pravděpodobnost, že





vývojový tým splní cíl sprintu. Vývojový tým se často ihned po Daily Scrumu schází a přeplánovává práci zbývající k dokončení Sprintu. Vývojový tým by měl být každodenně schopen vlastníkově produktu a Scrum masterovi vyložit, jakým způsobem se chce společně sebe-organizovat pro dosažení cíle a pro vytvoření očekávaného přírůstku ve zbývající části Sprintu.

Scrum master zajišťuje konání této schůzky, ale za průběh Daily Scrumu je zodpovědný samotný tým. Dále Scrum master vývojovému týmu pomáhá udržet Daily Scrum v 15-ti minutovém rámci.

Scrum master dbá na dodržování pravidla, že Daily Scrumu se účastní jen členové vývojového týmu. Daily Scrum není status meetingem, ale je tu pro pracovníky, kteří transformují položky produktového backlogu do přírůstku produktu.

Daily Scrum zlepšuje komunikaci, eliminuje potřebu dalších schůzek, identifikuje a odstraňuje překážky vývoje, vyzdvihuje a podporuje rychlá rozhodnutí a zlepšuje úroveň znalostí každého člena týmu o projektu. Je klíčovou schůzkou zajišťující kontrolu a adaptaci.

### Vyhodnocení sprintu (Sprint Review)

Vyhodnocení prováděné na konci Sprintu zkoumá přírůstek a v případě potřeby adaptuje produktový backlog. Během vyhodnocení sprintu Scrum tým a ostatní zúčastněné strany (stakeholders) probírají výsledky Sprintu. Dle zjištěných skutečností a také dle změn produktového backlogu (provedených během Sprintu) všichni zúčastnění spolupracují na rozhodnutích „jak postupovat dále“. Jde o neformální schůzku; předvedení přírůstku má za cíl získat zpětnou vazbu a podpořit spolupráci.

Při jednoměsíčních Sprintech by tato schůzka měla zabrat nejvýše čtyři hodiny. U kratších Sprintů by na vyhodnocení měl být vymezen úměrně kratší čas. Například dvoutýdenní sprinty budou mít dvouhodinové vyhodnocení.

Vyhodnocení sprintu má následující části:

- Vlastník produktu identifikuje, co bylo dokončeno („hotovo“) a co ne.
- Vývojový tým diskutuje o tom, co se během sprintu dařilo, k jakým problémům docházelo a jak tyto problémy byly řešeny.
- Vývojový tým demonstruje „hotové“ výsledky a odpovídá na dotazy týkající se přírůstku.
- Vlastník produktu diskutuje o aktuálním stavu produktového backlogu. Na základě dosaženého pokroku odhaduje pravděpodobné datum dokončení.
- A celá skupina společně navrhne, co bude řešeno dále, takže schůzka pro vyhodnocení sprintu poskytne důležitý vstup pro následující plánovací schůzku.

Výsledkem vyhodnocení sprintu je revidovaný produktový backlog, a dále jeho vytipované položky jako kandidáti na zařazení do dalšího sprintu. Produktový backlog může být také celkově přizpůsobený novým příležitostem.

### Retrospektiva sprintu

Scrum tým má během retrospektivy sprintu příležitost ke kontrole sama sebe a k naplánování kroků, pomocí kterých dojde během příštího sprintu ke zlepšení.





Retrospektiva následuje po vyhodnocení sprintu, ale ještě před plánovací schůzkou. Při jednoměsíčních sprintech by měla trvat tři hodiny. U kratších sprintů by měla zabrat přiměřeně kratší dobu.

Smyslem retrospektivy sprintu je:

- Kontrola toho, jak probíhal poslední sprint s ohledem na lidi, vztahy, procesy a použité nástroje.
- Identifikace a seřazení hlavních aspektů, které fungovaly dobře a které je možné zlepšit.
- Vytvoření plánu na zavedení jednotlivých vylepšení; tento plán je vytvářen stejným způsobem, kterým Scrum tým provádí svou běžnou práci při vývoji produktu.

Scrum master povzbuzuje Scrum tým ke změnám (myšleno ke změnám v rámci procesu Scrumu) – ke zlepšení vývojového procesu a postupů za účelem efektivnějšího a příjemnějšího průběhu příštího sprintu. Během každé retrospektivy Scrum tým uvažuje o tom, jak vhodnou adaptací definice „hotovo“ zvýšit kvalitu produktu.

Scrum tým by měl mít na konci retrospektivy hotový seznam zlepšení, která v dalším Sprintu zavede. Zavádění těchto zlepšení v dalším Sprintu je vlastně adaptací plynoucí z kontroly, kterou Scrum tým provedl sám na sobě. I když zlepšení mohou být zaváděna kdykoliv, retrospektiva sprintu přináší týmu metodickou a formalizovanou příležitost k zaměření se na kontrolu a adaptaci.

## Artefakty

Artefakty Scrumu reprezentují práci a její hodnoty různými způsoby, které jsou užitečné pro poskytování transparentnosti a dále jsou příležitostí pro kontrolu a adaptaci. Artefakty definované Scrumem jsou navrženy tak, aby maximalizovaly transparentnost klíčových informací potřebných k zajištění toho, aby Scrum tým byl úspěšný v dodání hotového přírůstku.

## Produktový backlog

Produktový backlog je prioritizovaný seznam všeho, co může být potřeba v produktu a je jediným zdrojem požadavků na jakoukoliv změnu v produktu. Vlastník produktu je zodpovědný za obsah, dostupnost a prioritizaci backlogu. Produktový backlog není nikdy úplný. V úvodní fázi vývoje obsahuje pouze zpočátku známé a dobře pochopené požadavky. Produktový backlog se vyvíjí tak, jak se vyvíjí produkt a prostředí, ve kterém bude používán. Produktový backlog je dynamický, stále se mění tak, aby obsahoval to, co produkt potřebuje, aby byl vhodný, konkurenceschopný a užitečný. Dokud existuje produkt, existuje produktový backlog.

Produktový backlog je seznam všech vlastností, funkcí, požadavků, rozšíření a oprav chyb, které představují všechny změny, které budou provedeny v produktu v příštích vydáních. Položky produktového backlogu mají popis, prioritu a odhad.

Produktový backlog je často prioritizován podle přínosu, míry rizikovosti, priority a nezbytnosti. Položky produktového backlogu s nejvyšší prioritou řídí okamžité vývojové aktivity. Čím vyšší priorita,







tím více se o ní přemýšlelo a tím více je shoda ohledně její hodnoty. Položky backlogu s vyšší prioritou jsou jasnější a mají více detailních informací, než položky backlogu s nižší prioritou. Díky tomu jsou pro ně k dispozici přesnější odhady. Čím nižší priorita, tím méně detailů. Položky produktového backlogu, které zaměstnají vývojový tým v nadcházejícím Sprintu, jsou rozpracovány a rozděleny tak, že každá položka může být dokončena ve stavu „hotovo“ během jednoho sprintu. Položky Produktového backlogu, které mohou být dodány ve stavu „hotovo“ v rámci jednoho sprintu, jsou považovány za připravené a vhodné pro výběr na plánovací schůzce. Jak je produkt používán a jeho hodnota roste, trh poskytuje zpětnou vazbu a produktový backlog se vyvíjí do rozsáhlejšího a podrobnějšího seznamu. Požadavky na změny nikdy nepřestanou. Produktový backlog je živý dokument. Změny v byznys požadavcích, podmínkách trhu a technologiích mohou způsobit změny v produktovém backlogu.

Často pracuje společně na stejném produktu více scrum týmů. V takovém případě se používá pouze jeden produktový backlog a jeho položky se seskupí podle nějakého atributu.

Péče o produktový backlog zahrnuje zpřesňování detailů, odhadů pracnosti a uspořádání jednotlivých položek v něm. Je to stále probíhající proces detailního zpřesňování jednotlivých položek produktového backlogu při němž spolupracuje vlastník produktu a vývojový tým. Během přípravy produktového backlogu jsou odhady přezkoumávány a revidovány. Nicméně mohou být aktualizovány kdykoliv vlastníkem produktu na základě jeho vlastního uvážení.

Během sprintu je příprava produktového backlogu práce na částečný úvazek pro vlastníka produktu a vývojový tým. Často má sám vývojový tým nejlepší znalost, jak provádět přípravu. Jak a kdy je příprava hotova, je na rozhodnutí Scrum týmu. Příprava obvykle nezabere více než 10% kapacity vývojového týmu. Vývojový tým je odpovědný za všechny odhady. Vlastník produktu může ovlivnit vývojový tým tím, že mu pomůže porozumět a vybrat kompromisy, ale finální odhad je vytvořen vývojovým týmem, který bude provádět odhadovanou práci.

### Sledování procesu směřujícího k dosažení cíle

Celková práce zbývající k dosažení cíle může být kdykoliv vyčíslena. Vlastník produktu sleduje sumu zbývající práce minimálně na každém vyhodnocení sprintu. Vlastník produktu porovnává toto množství práce s množstvím zbylé práce z předchozích vyhodnocení sprintů tak, aby posoudil postup plánovaných prací směřující k cíli v požadovaném čase. Tato informace je srozumitelná a viditelná pro všechny zúčastněné strany.

K odhadování budoucího průběhu projektu se používají grafy zachycující trendy (burndown, burnup), jakož i jiné prediktivní techniky. Byť jsou tyto grafy jistě užitečné, nemohou nahradit důležitost znalosti založené na zkušenostech. Ve složitých prostředích je těžké odhadnout, co se stane. Jedině to, co se již stalo, může být použito pro rozhodování o budoucím vývoji.





## Sprint Backlog

Sprint backlog je množina úkolů vybraných z produktového backlogu včetně plánu dodání produktového přírůstku a splnění cíle sprintu. Sprint backlog je prognóza vývojového týmu jaká funkcionality bude dodána v následujícím přírůstku a kolik práce bude potřebné k dodání této funkcionality.

Sprint backlog definuje práci vývojového týmu, která je potřebná k tomu, aby tým dodal položky produktového backlogu do hotového přírůstku a splnil tak cíl daného sprintu.

Sprint backlog je natolik podrobný, že změny jeho průběhu mohou být sledovány na denní bázi, tj. na Daily Scrumu. Sprint backlog se během sprintu mění; vývojový tým ho upravuje po celou dobu sprintu. Tyto úpravy jsou potřebné proto, že vývojový tým během sprintu získává více poznatků o tom, co je potřeba ke splnění cíle sprintu. Pokud je požadována další práce, vývojový tým ji přidá do sprint backlogu. Podle toho, kolik je na úkolech odpracováno, nebo jestliže jsou dokončeny, provádí se aktualizace odhadu. Pokud se některé úkoly ukáží jako zbytečné, jsou z backlogu odstraněny. Jedině vývojový tým může změnit sprint backlog v průběhu sprintu. Sprint backlog tak představuje jasný, transparentní a aktuální obraz práce, kterou vývojový tým plánuje v průběhu sprintu dokončit. Sprint backlog je výlučně záležitostí vývojovému týmu.

### Sledování průběhu sprintu

Množství práce, která zbývá k dokončení sprintu, může být vypočteno v kterémkoli okamžiku. Vývojový tým sleduje toto množství práce minimálně na denní bázi (Daily Scrum) a předpovídá pravděpodobnost dosažení cíle sprintu. Vývojový tým řídí svůj postup během sprintu sledováním množství zbývajících práce.

Scrum nebere ohled na čas strávený prací na úkolech sprint backlogu. Středem pozornosti je pouze zbývajících práce a termín.

## Přírůstek

Přírůstek je suma všech dokončených položek produktového backlogu v průběhu aktuálního sprintu a všech předchozích sprintů. Koncem sprintu musí být nový přírůstek „hotový“, což znamená, že musí splňovat podmínky definované Scrum týmem pro stav „hotovo“ a také musí splňovat podmínky použitelnosti bez ohledu na to, jestli se rozhodne vlastník produktu ho vydat nebo ne.

### Definice toho co je „hotovo“

Když je položka produktového backlogu anebo přírůstek produktu prohlášen za „hotový“, každý musí rozumět, co znamená „hotovo“. Tato definice se mezi Scrum týmy výrazně liší. Členové jednoho týmu





musí ovšem vnímat práci jako dokončenou stejným způsobem. Proto má Scrum tým definici „hotovo“, která se používá k posouzení, zda je práce na přírůstku produktu dokončena.

Definice „hotovo“ také vede vývojový tým k poznání, kolik položek může přijmout do sprintu v průběhu plánovacího schůzky pro sprint. Cílem každého sprintu je dodání přírůstku potenciálně nasaditelného produktu, a který je plně v souladu s definicí „hotovo“ Scrum týmu. Vývojové týmy dodávají přírůstek funkčnosti produktu v každém sprintu. Tento přírůstek je použitelný, takže vlastník produktu se může rozhodnout k jeho okamžitému nasazení. Každý přírůstek je doplňkem ke všem předcházejícím přírůstkům a je intenzivně testován, čímž se zajistí, že všechny přírůstky spolu spolupracují. Jak Scrum tým vyžívá, očekává se, že jeho definice „hotovo“ se bude rozšiřovat tak, aby pokryla přísnější kritéria pro vyšší kvalitu.

## Závěr

Scrum nabízíme bezplatně v tomto průvodci. Role ve Scrumu, artefakty, schůzky a pravidla jsou neměnné. I když je možné implementovat Scrum jenom z části, výsledek není Scrum. Scrum existuje pouze v celku a slouží jako kontejner pro další techniky, metodiky a postupy.

## Poděkování

### Lidé

Z tisíců lidí, kteří přispěli ke vzniku Scrumu, bychom chtěli připomenout ty, kteří významně pomohli v prvních deseti letech. Jako první to byli Jeff Sutherland, který spolupracoval s Jeffem McKennou, a Ken Schwaber s Mikem Smithem a Chrisem Martinem. V dalších letech pomáhalo mnoho dalších lidí, a bez jejich pomoci by Scrum nebyl tak vylepšen, jako je tomu dnes. Při formulování této verze Scrum Guidu poskytli své editační zručnosti David Starr.

## Historie

Ken Schwaber a Jeff Sutherland jako první spolu prezentovali Scrum na konferenci OOPSLA 1995. Tato prezentace v zásadě dokumentovala poznání, které Ken a Jeff nabyli v průběhu předcházejících let při aplikování Scrumu. Scrum se vyvíjí už delší dobu. Pokud bychom chtěli připomenout místa, kde byl poprvé vyzkoušen a zdokonalen, musíme zmínit firmy Individual, Inc., Fidelity Investments a IDX (dnes GE Medical).

