

SOFTWAREVÉ INŽENÝRSTVÍ

SI

Řízení IT projektů

Ing. Ondřej Macek

2013/14



ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

Jak vypadal vývoj SW?

- Bylo třeba specifikovat zadání, to se naprogramovalo a pak se dostal výsledek – podobně jako např. ve strojírenství
- Zákazník nerozuměl tvorbě programů (magie) – dnes si všichni myslí, že rozumí
- Přelom 81 – první PC, takže bylo třeba začít myslet jinak. Jde to těžko!
- Někdy kolem 78 byla konference NATO – o softwarovém inženýrství – projekty často chybují, nejsou včas, ...
- V roce 2000 bylo 125 aplikací pro řízení projektů a 25 technik – a to ještě nebyli agilní metodiky

S

I

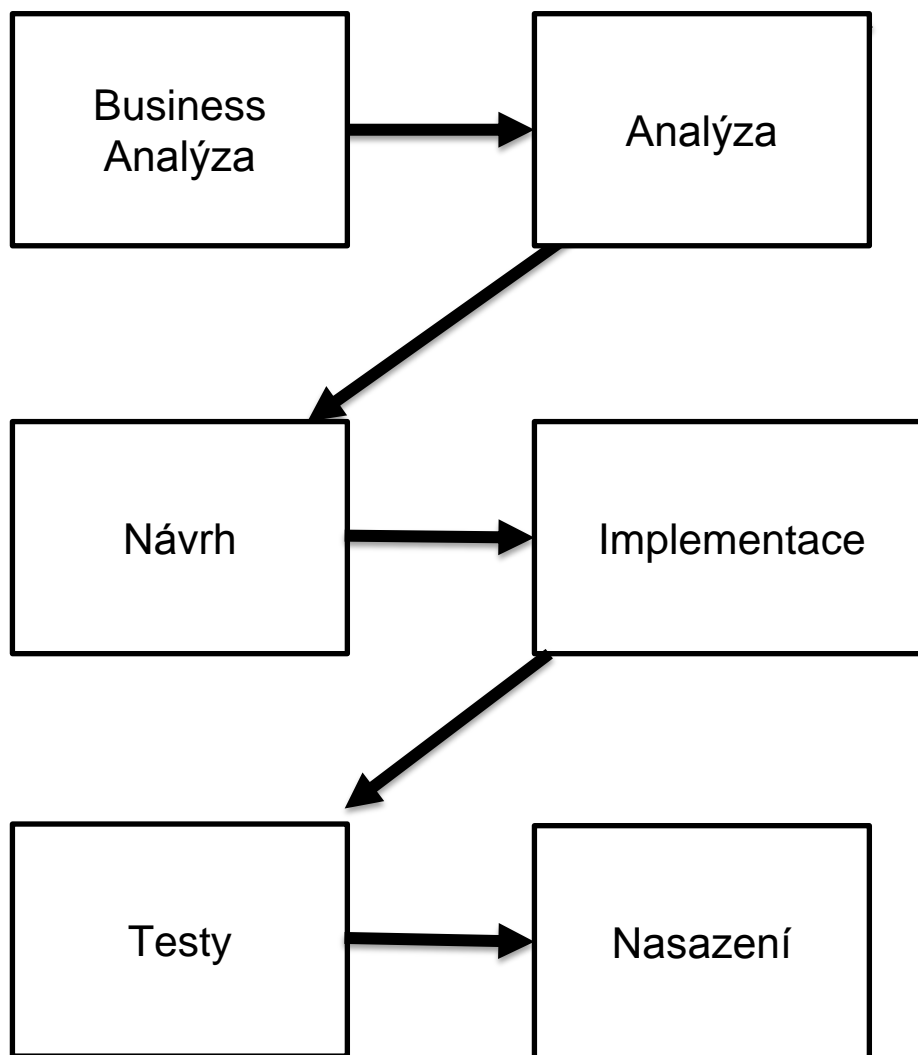
Tradiční řízení

Tradiční řízení

Tradiční řízení

Tradiční řízení

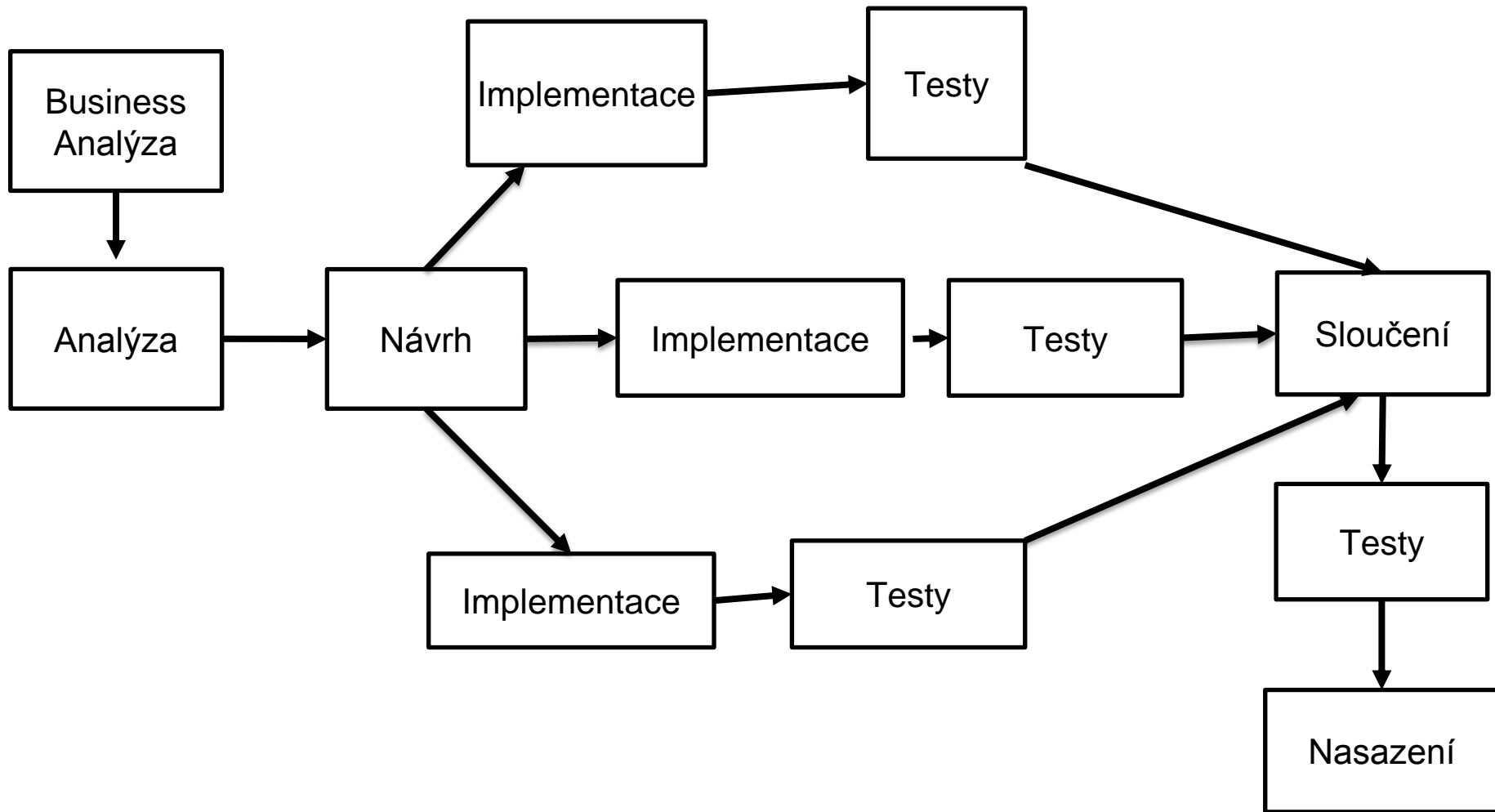
Lineární přístup k řízení projektů (LPMLC)



- Silné stránky?
 - Už od začátku je jasný plán => víme co bude
 - snadné řízení zdrojů (paralelismus projektů)
 - Zvládne to i začátečník/průměrný zaměstnanec (rozdíl oproti agilním)
 - Není třeba aby tým byl na jednom místě – analýza u nás, implementace v Malajsii; dá se snadno předat externímu dodavateli

- Slabé stránky?
 - Špatně reaguje na změnu
 - Stojí hodně peněz – veškeré chyby se projeví až při předání (akceptačním testu) a pak je drahé chyby opravit
 - Dlouho trvá než zákazník vidí nějaký výstup
 - Vyžaduje detailní plán a specifikaci – každé přeplánování je drahé
 - Sleduje procesy, které není možné změnit
 - Není zaměřen na přínos klientovi – je zaměřen na dodání v nějaký čas – dodání podle specifikace není často totožné s tím co zákazníkovi pomůže – potřeby vs.přání

Rapid LPMLC



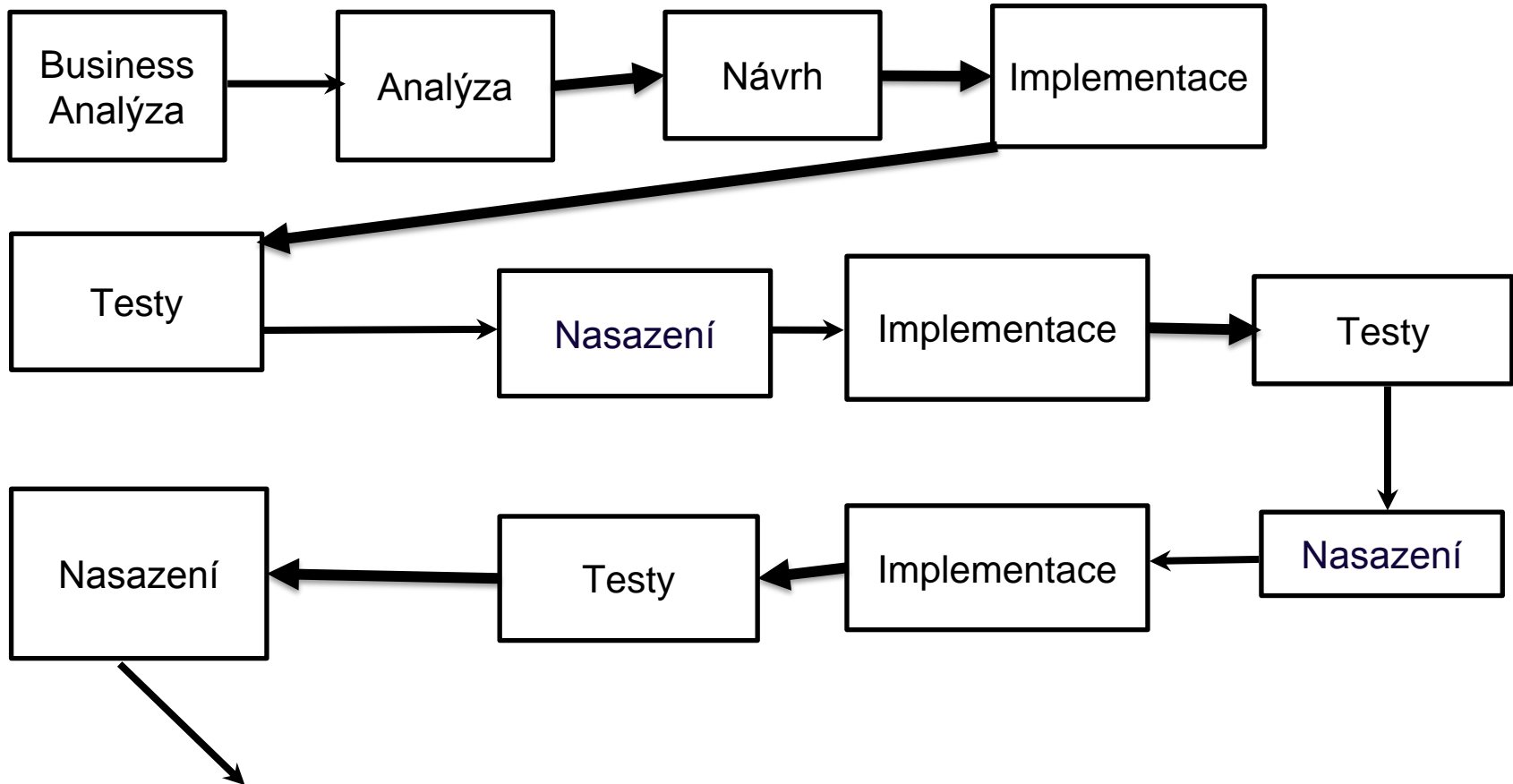
- Silné stránky?
- Slabé stránky?

The logo features a large, bold, orange letter 'S' on the left. To its right is a vertical orange bar, followed by a horizontal grey bar. The text 'Pokročilejší přístupy' is written in white on the grey bar.

S

Pokročilejší přístupy

Inkrementální přístup



Inkrementální přístup

- Silné stránky?
- Slabé stránky?

- Rápidní inkrementální přístup

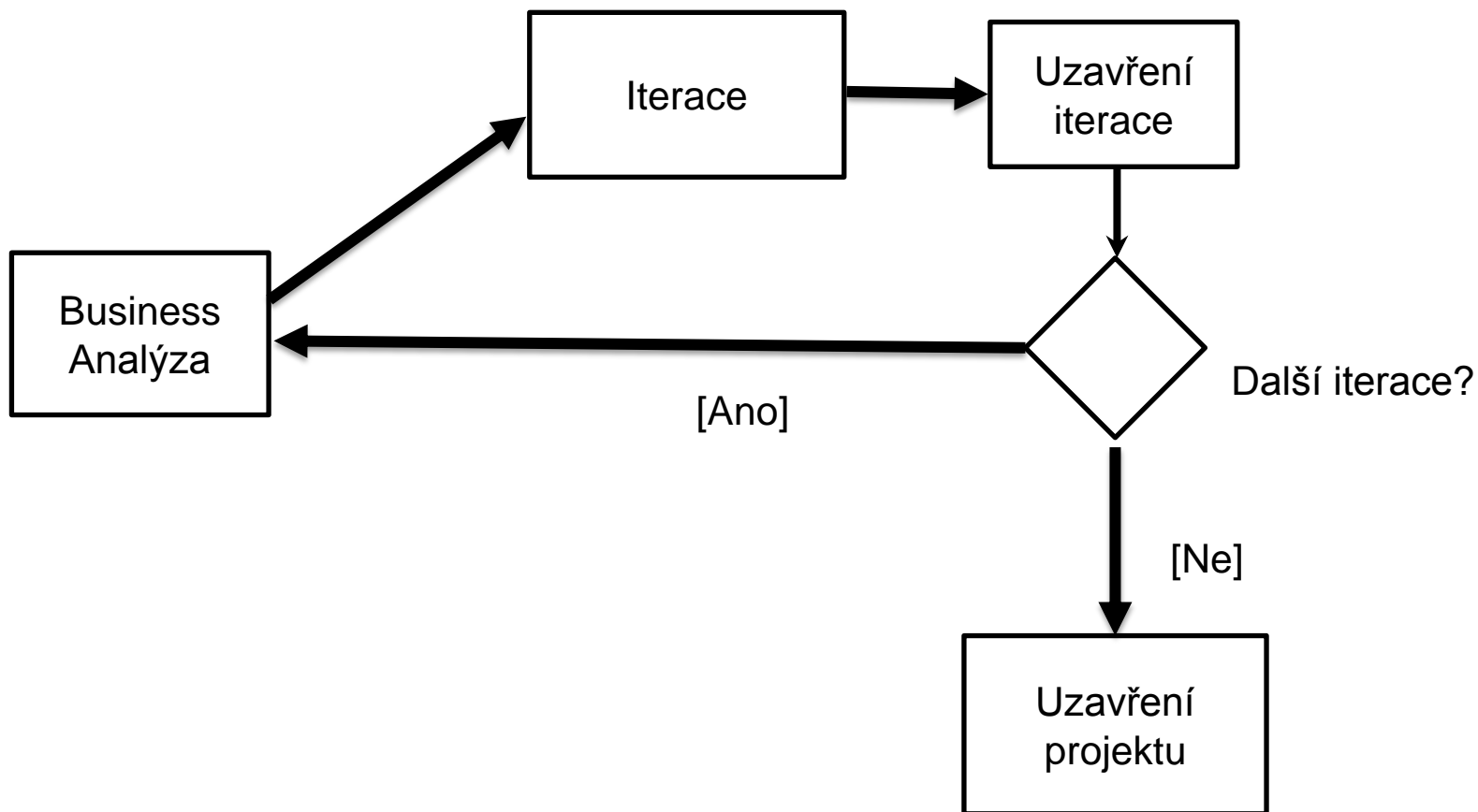
- Silné stránky?
- Slabé stránky?

S

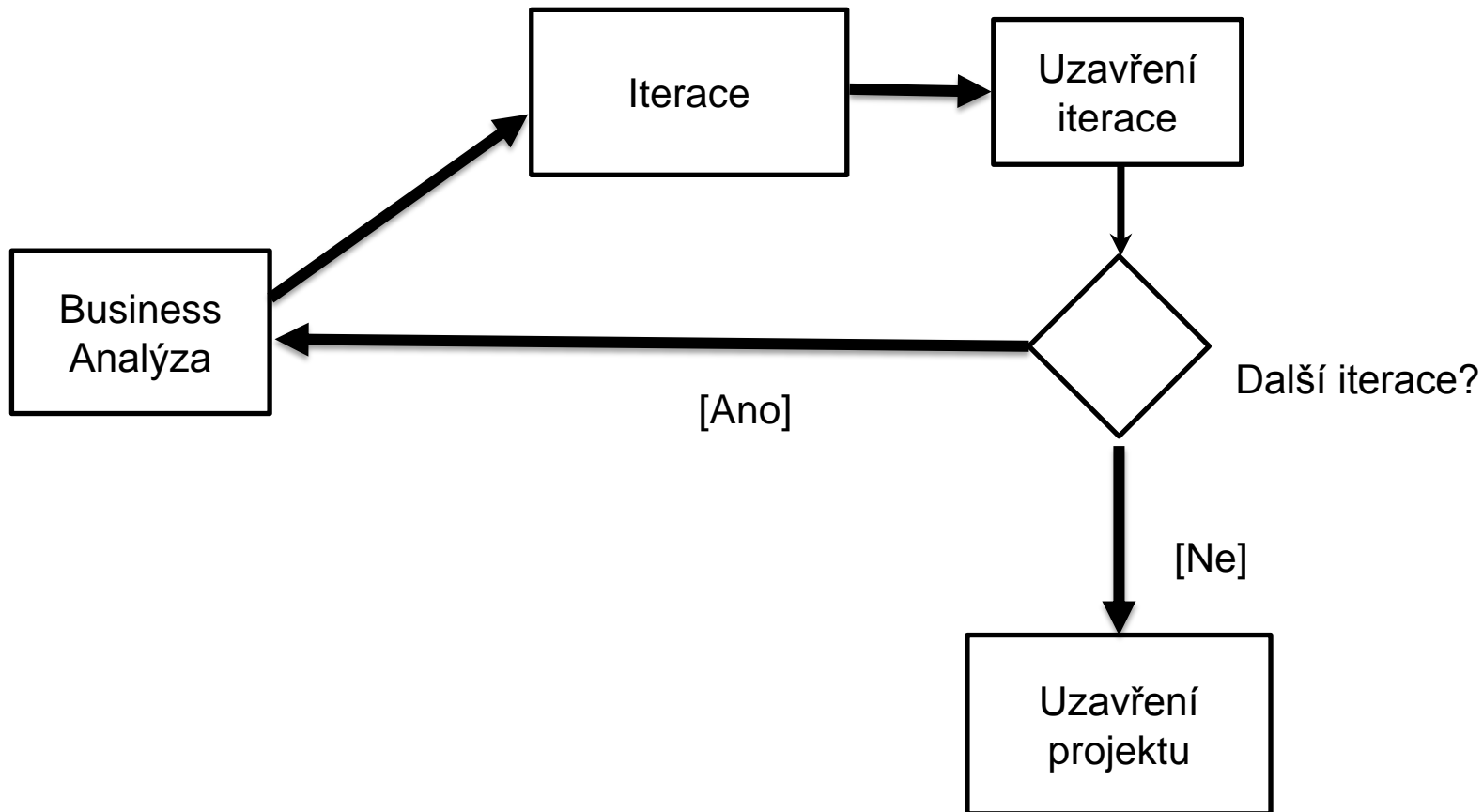
I

Iterační řízení

Iterační přístup



Co je iterace?



Charakter

- Častý sběr požadavků na změnu, ale známe řešení
- Tým v jedné lokalitě
- Iterace 2-4 týdny
- Část řešení není známa – nejsou detaily -> změny
- High level scope/detailní scope pro každou iteraci
- Zpětná vazba s klientem
- Jednoduché/lehké zapojení klienta

Nevýhody

- Zapojení klienta
- Týmy v jedné lokalitě
- Není vždy jasné jak to dopadne

Potřeby

- Malý tým
- Zkušení pracovníci

S

I

Unified Process

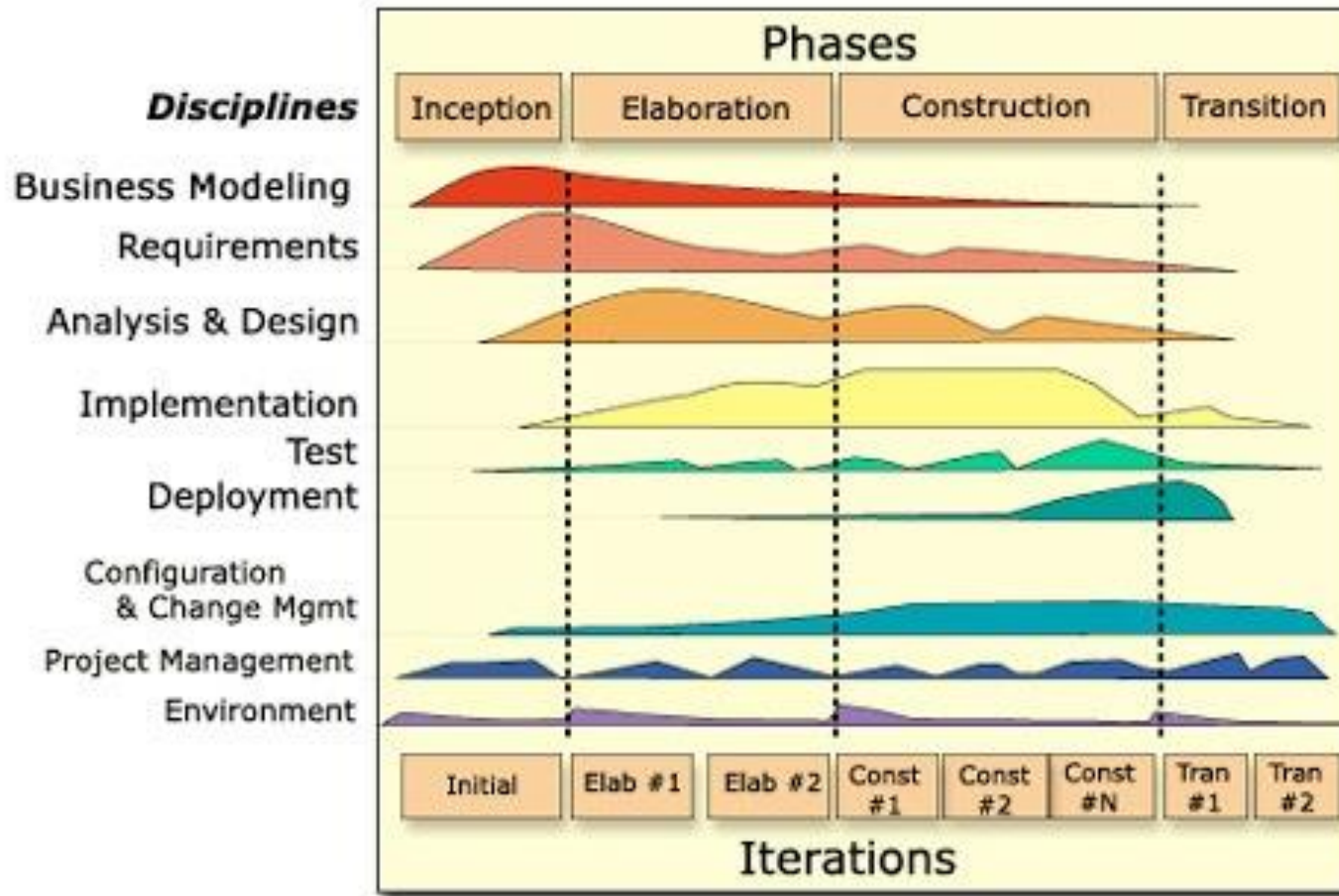
Unified Process

Unified Process

Unified Process

- Postupuje ve fázích
- Fáze se skládá z iterací
- Po dokončení každé iterace/fáze se provede zhodnocení
- Další iterace/fáze projektu začne jen v případě splnění kritérií.

FÁZE RUP



ZAHÁJENÍ (INCEPCE)

- identifikace problému, který má být vyřešen
- formulace čeho má být dosaženo – specifikace zadání
- je třeba porozumět doméně
- určení priorit

Vstupy

Stávající systém, úvodní požadavky na systém,...

Výstupy

Vize, Plán, Základní požadavky, (Prototyp)

Milník

Dohoda mezi zákazníkem a dodavatelem na rozsahu, ceně a harmonogramu

PŘÍPRAVA (ELABORACE)

- sestavení projekčního týmu
- nastavení prostředí pro vývoj
- zpřesňování představy/požadavků
- základ architektury a prototypu
- plán projektu

Výstup

Prototyp, Use case (80%), Návrh(10%), Datový model, Plán testů, Plán projektu,

Milník

Architektura, Use case, Plán testů, Podrobnější plán projektu

- dokončení návrhu
- první verze SW
- návrh testů
- realizace testů
- vznik dokumentace (návrhové modely, model nasazení...)

Výstupy

První verze, testy, uživatelská dokumentace

Milník

Funkční verze

PŘEDÁNÍ (TRANSITION)

- předání finální verze koncovým uživatelům
- beta testování systému
- nemělo by docházet k zásadním změnám funkcionality
- řeší se instalace, konfigurace a odladování

Výstupy

Konečná funkční verze splňující všechny požadavky zadání, Manuály

Milník

Předání

- Vyžaduje konzultace s klienty
- Základem analýzy a implementace jsou use-cases
- Na architekturu je třeba zaměřit se co nejdříve
- Testy probíhají na všech úrovních

S

I

AGILNÍ PRINCIPY

- Metodiky vývoje SW jsou nevhodné
- Špatná komunikace vede ke krachu projektu
- Na testy už nezbyvá čas
- Vytváření „zbytečnou“ dokumentaci

Agilní přístup - motivace

- Spokojený zákazník
- Spokojený programátor

- způsob řízení týmu
- způsob vývoje SW
- způsob myšlení

- **Člověk a komunikace je nad procesy a nástroje**

- **Člověk a komunikace je nad procesy a nástroje**
- **Fungující software je nad vyčerpávající dokumentací**

- **Člověk a komunikace je nad procesy a nástroje**
- **Fungující software je nad vyčerpávající dokumentací**
- **Spolupráce se zákazníkem je nad přesně sjednanou smlouvou**

- **Člověk a komunikace je nad procesy a nástroje**
- **Fungující software je nad vyčerpávající dokumentaci**
- **Spolupráce se zákazníkem je nad přesně sjednanou smlouvou**
- **Reakce na změnu je nad přesné dodržování plánu**

Role projektového manažera



S

I

Programování

Programování

Programování

Extrémní programování XP

- Projekt se skládá z **iterací**
- Plánování je zaměřeno na **časté dodání malých releasů** (iterace = release)
- Každý tým má vyčleněný **vlastní pracovní prost**
- Pracovní den začíná **stand up meetingem**
- Reporting sleduje **rychlost projektu**
- Popis funkce **pomocí User stories**.

- Návrh je **jednoduchý**
 - (Testable, Understandable, Browsable, and Explainable)
- Použití **metafor**
- Použití CRC při návrhových meetinzích (Class, Responsibilities, and Collaboration)
- **Prototypování** SW i dílčích částí
- Zaměření na **okamžitou přidanou hodnotu**
- **Refactoring**
- **Zákazník je vždy k dispozici**
- **Coding standards**
- **Test driven development**
- **Pair programming**
- **Continuous integration**
- **Sdílený kód**

- **Pro každý kód existuje unit test**
- **Před každým release** musí projít **všechny testy**
- Pokud je objeven **bug => je napsán unit a akceptační test**
- Akceptační testy probíhají často a jejich úspěšnost je zveřejňována

S

I

FRAMEWORK

FOR

SCRUM

Zúčastněné strany

- Product owner
- Scrum master
- Team

SCRUM

dokumenty/výstupy

- Sprint/Product Backlog
- Burndown charts
- Increment

Project/Sprint Backlog

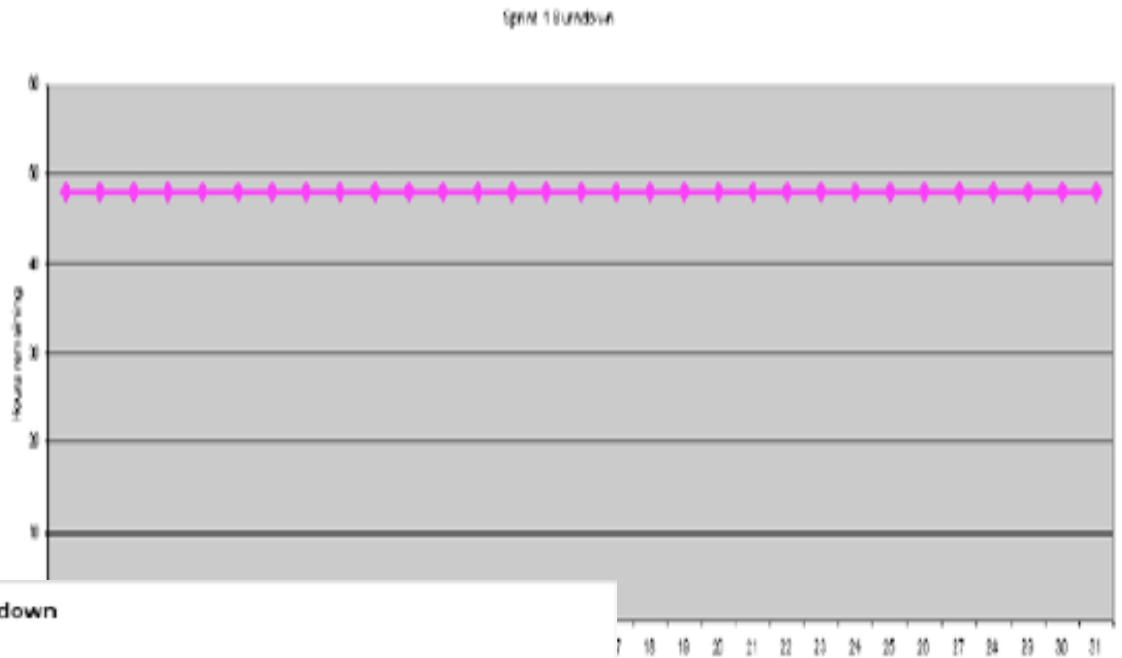
Project

Sprint	Name	Estimate (days)	Remaining (days)
1	Manažer zadává úkoly zaměstnancům	1	1
	Zaměstnanec předá úkol pro otestování	2	2

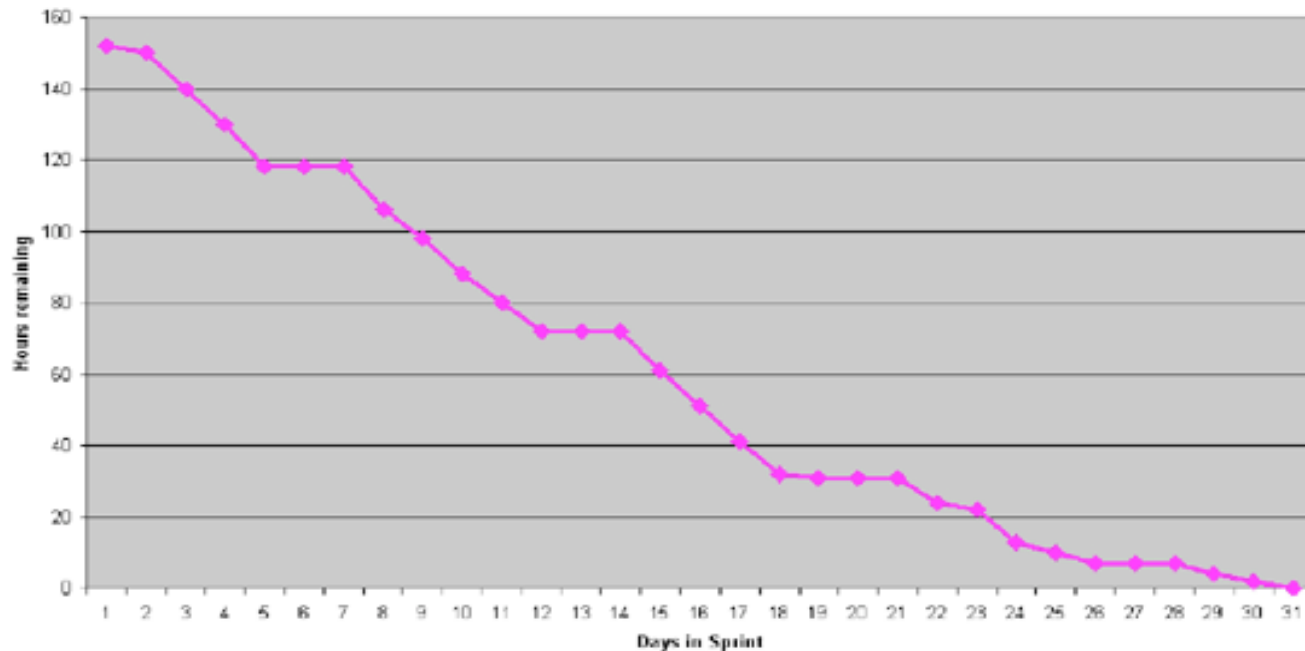
Sprint

Name	Estimate (hours)	1. 12.	2. 12.	3. 12.
Manažer zadává úkoly zaměstnancům	8	7	3	0

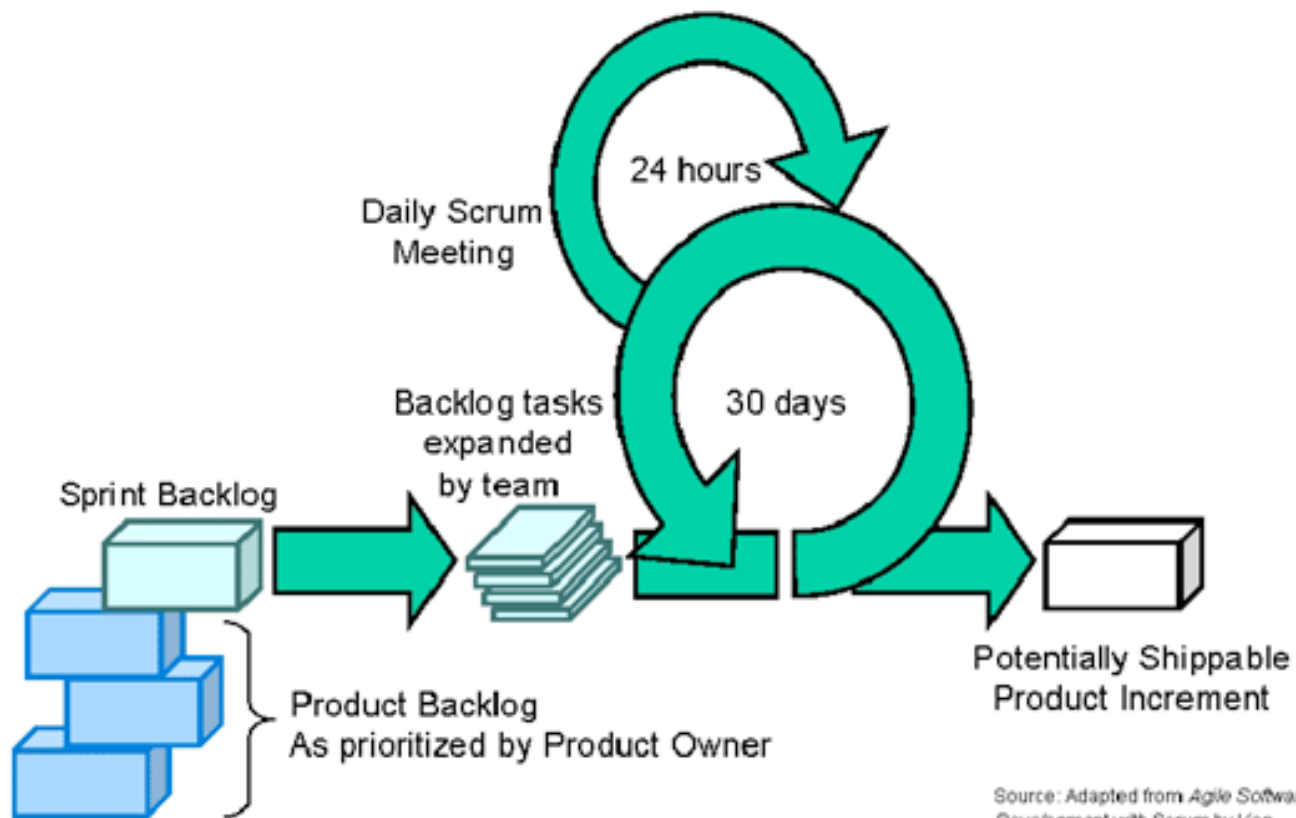
Burndown graph



Sprint 1 Burndown



Životní cyklus



Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

SCRUM Meetings

- Planning meeting
- Daily meetings
- Sprint review
- Sprint retrospective meeting

- Kompletní funkčnost
- Kompletní testy
- Připraveno na nasazení

S

I

1

2

3

KANBAN

Co je kanban?

看板

- Signální karta
- Vizuální metoda řízení a zlepšení projektů/výroby
- Cílem je efektivita dodávky a práce týmu
- Efektivita je založená na znalosti
 - co produkovat
 - v jakém množství produkovat
 - kdy dodat

1. Začni s tím co umíš
2. Začni měnit své postupy a role (inkrementální evoluce)
3. Respektuj své “kořeny”
4. Leadership na všech úrovních

1. Vizualizace workflow
2. Omez work-in-progres
3. Řid' tok (manage workflow)
4. Vysvětli principy fungování organizace (policies)
5. Uč se
6. Zlepšuj a experimentuj

Kanban board



S

I

LEAN DEVELOPMENT

LEAN DEVELOPMENT

LEAN DEVELOPMENT

1. Eliminace odpadu
 - nedělej nic co nepřináší hodnotu
 - suffer oriented programming
2. Soustavné učení
 - krátké iterace + feedback
3. Rozhodování na poslední chvíli
 - když je dostatek informací
4. Dodávání co nejrychleji
5. Posílení role týmu
6. Zapracování integrity
 - zákaznická
 - systémová
7. Soustředění se na celek

S

I

Agile

Výhody a problémy (s) Agile

- Vize
- Týmový práce & spolupráce
- Jednoduchá pravidla
- Sdílení informací
- Lehké řízení
- Učení se

Kde se Agilní přístup doporučuje

- Zkušení programátoři
- Očekává se hodně změn
- Malý tým

Co Agilní přístup potřebuje

co je doporučováno

- Zkušené programátory
- Někoho s předchozí zkušeností s Agile
- Komunikační schopnosti

Kde se Agilní přístup nedoporučuje

- Velké programátorské týmy
- Dislokované týmy
- Nutnost fixace projektového trojúhelníku

Co Agilní programování není

- Anarchie
- Ignorování dokumentace
- Cowboy programming
- Striktní sada pravidel

Problémy s aplikací agilního přístupu

- Strach
- Neznalost
- Firmy jsou řízeny procesně
- Manažeři
- Programátoři

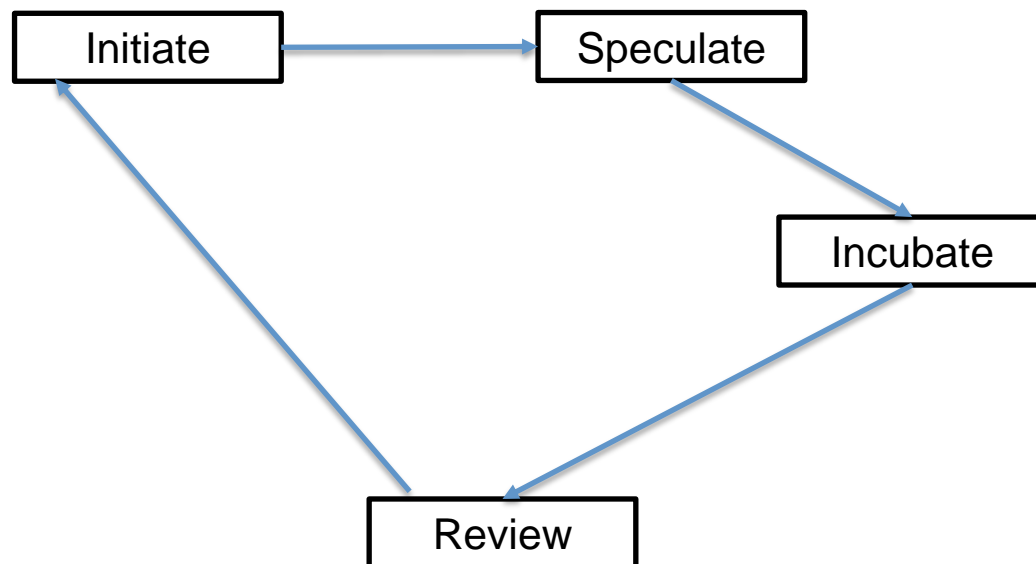
- Aplikace jen některých doporučení
- Nepřítomnost zákazníka
- Spoléhá na ústní podání
- Přečtení knížky \neq zavedení do praxe
- Malý důraz na architekturu v počátečních fázích

The logo features a large, bold, orange letter 'S' on the left. To its right is a vertical orange bar, followed by a horizontal grey bar. The text 'Extrémní řízení projektů' is written in white on the grey bar.

S

Extrémní řízení projektů

XŘP – cyklus projektu



- <http://soch.cz/blog/>
- <https://docs.google.com/spreadsheet/ccc?key=0Ag91X5gS0piQcDhvTUw1QzINLS11d29mbWhsMW5wZXc#gid=0>
- <http://www.extremeprogramming.org/>
- <http://www.scrumalliance.org/>
- <http://www.methodsandtools.com/archive/archive.php?id=18>