

SI2 – BI-WSI-SI-24

Projektové řízení a měření: modely SDLC, plánování krátkodobé a dlouhodobé, kategorie metrik a jejich využití, historie projektu, řízení rizik, odhady, způsob jejich tvorby a verifikace

Obsah

1	SDCL - Software development lifecycle	2
1.1	Agile vs Plan-Driven	2
1.2	Waterfall - dlouhodobé plánování	2
1.3	Iterativní - středně dlouhé plánování	3
1.4	Agilní - krátkodobé plánování	3
2	Metriky	4
2.1	Historie	4
2.2	Odhady, jejich tvorba a verifikace	4
3	Řízení rizik	4

1 SDCL - Software development lifecycle

Jedná se o množinu aktivit nutných k tomu, aby SW vznikl. SDLC definuje vztahy, prostřední, přístup, čas a jakékoli jiné vlastnosti primárních a podpůrných činností SW inženýrství.

1.1 Agile vs Plan-Driven

Příklady modelů životního cyklu:

- waterfall
- iterativní model
- SCRUM

Aktivity se rozdělují na Primární a Podpůrné

- primární
 - business modeling
 - sběr požadavků
 - analýza a návrh
 - programování
 - testování
 - nasazení
- podpurné
 - projektové řízení
 - správa prostředí
 - změnové a konfigurační řízení

Primární činnosti "tvoří hodnoty", probíhají do určité míry sériově (za sebou), většinou i v opakujících se cyklech.

Podpůrné činnosti probíhají stále, po celou dobu celého projektu a primární činnosti "zaobalují".

1.2 Waterfall - dlouhodobé plánování

Oddělené fáze:

- analýza požadavků
- design
- implementace
- testování
- provoz

Výhody:

- jasně definovaný plán
- predikovatelnost (čas, rozsah, cena)
- snadná koordinace práce

Nevýhody:

- nutno chápat, co se chce již na začátku
- reakce na změny (požadavky, termíny)
- rychlost dodávky (kdy nejdříve uvidí klient nějaký progress)
- integrace více systémů

1.3 Iterativní - středně dlouhé plánování

Změny oproti vodopádu:

- několik verzí systému
- jednotlivé verze se dělají vodopádem

Výhody - stejné jako vodopád, jen má navíc :

- zákazník má přístup k verzím/prototypům (vidí co dostává)

Nevýhody

- nutno chápat, co se chce od začátku, možné změny do dalších verzí

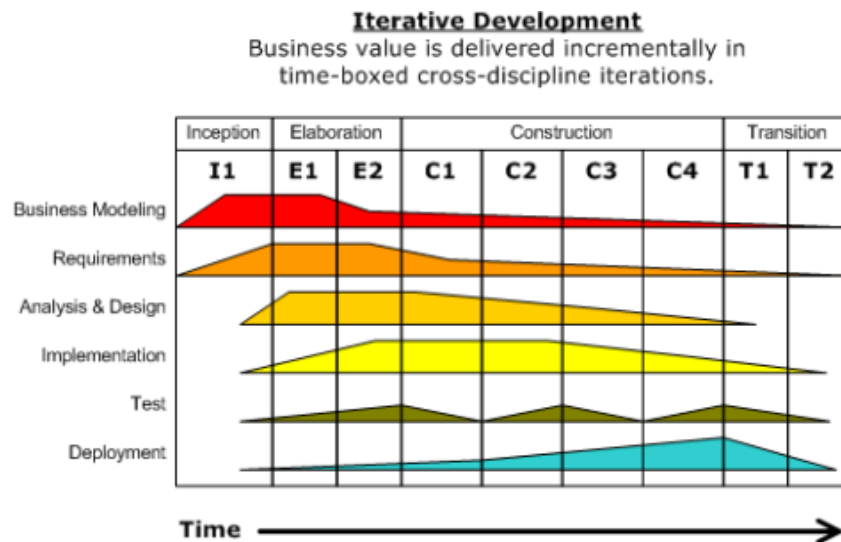


Figure 1: Iterativní vývoj

1.4 Agilní - krátkodobé plánování

Změny oproti iterativnímu:

- mnohem kratší iterace
- jednotlivé verze ne vždy produkční
- velké nároky na celý tým
- změna myšlení

Výhody:

- rychlejší vývoj nové verze
- rychlejší reakce na zpětnou vazbu

- nové funkce se objeví dříve

Nevýhody:

- nutné kontinuální zapojení všech členů týmu
- nutný silný business vlasník

2 Metriky

Základní metriky:

- Time (kalendářní čas - jak dlouho to bude trvat udělat)
- Size/Scope (jaký je rozsah)
 - počet řádků kódu
 - počet obrazovek
 - počet tříd
 - ...
- Effort (jak je to pracné - většinou udávané v MD)
- Quality (jakost - jak moc se hledí na kvalitu → výskyt chyb)

Pro zaznamenávání historie práce (projektů a jednotlivých úkolů) se používá Ticketovací systém (youtrack, github, redmine, ...). Díky tomu se lépe odhadne časová pracnost na základě minulých dat. Podle naměřených metrik se odvíjí nabídka a cena.

2.1 Historie

Dokument nebo systém, který obsahuje informace a metriky z jednotlivých realizovaných projektů. Na jejich základě se lépe odhadují budoucí projekt, zpřesňovat odhady nových projektů, ekonomická hlediska a náročnost práce/údržby.

2.2 Odhady, jejich tvorba a verifikace

Metody odhadů:

- Dekompozice zadání na elementární části
- odkad na základě historie
- odhad by měl být konzistentní a měl by být zkontrolován dalšími účastníky
- odhad může být proveden na základě metrik
- použití standartizovaných metodik, které pracují s historií
- lze realizovat pomocí checklistu (na které problémy se zaměřit)

3 Řízení rizik

Riziko je ohrožení projektu/ceny/termínu/kvality. Musí se o nich zákazník dozvědět co nejdříve, protože mohou hrát roli při uzavření smlouvy, ceny, odhadu pracnosti...U každého rizika je nutné určit:

- pravděpodobnost, že nastane
- případný odhad

- jeho stav
- plán na snížení negativních dopadů