

06 Projektové řízení (2h)

tags: řsss-základ

> Projektové řízení. Mezinárodní standardy projektového řízení. Projekt, program, portfolio (PPP). Životní cyklus IT projektů. Procesní skupiny v projektu. Plánování projektů. Výpočet nákladů projektu. Ganttovy diagramy a projektové milníky. Síťová analýza, metoda kritické cesty (CPM), Program Evaluation and Review Technique (PERT). Zajištění kvality projektu, testy, přezkoumání, měření a standardy. (PA179)

> **Na co se ptají:** > **Bühnová:** Teamwork, optimální množství členů v týmu, k čemu slouží Quality Assurance, homogenita vs diverzita členů v týmu; TOC, CC, Gant; WBS, Gant, síťový diagram, PERT, CPM, rizika (třeba úbytek zdrojů - lidí, fyzické zdroje). Project management obecně, digramy, rizika > > **Ráček:** PPP - co to je, definice; jednotlivé části project planningu; PPP, co se týká standardů; Srovnání IPMA, PMI, PRINCE2; Rozdíly program, portfolio, proces, projekt; PPP, co to je, expozice a analýza rizik. Postup analýzy rizik pomocí stromu, AOA/AON - výhody a nevýhody (AOA má prý výhodu, že se v něm dobře zobrazuje milestone právě v těch nodech); síťové diagramy; PERT, Gant; Typy modelu životního cyklu - chtěl vědět od začátku na typech projektu kde by se zvolil jaký typ, za jakých okolností; SCRUM a RUP metodu kde u RUP chtěl slyšet jednotlivé názvy etap. Rozdíl mezi iterativním a inkrementálním cyklem. Kvalita SW, trojúhelník kvality - jak měřit kvalitu, příklad udržitelnosti kvality + testování SW, v které etapě se používá jaký typ testování a k čemu nám jsou dobré, popsat systémový test a co obnáší + integrační test (jak ho implementovat a co má být cílem). Jak probíhá komisi kontrola v průběhu vývoje, kdo se jí účastní a na co se zaměřují. Cyklomatická složitost algoritmu, co to je a kdy se používá. Metriky, výhody/nevýhody LOC, funkční body - chtěl vědět, jak získat a co konkrétně reprezentují jednotlivé části systému (EI, EO, EQ, ILF, EIF); Unified Process (fáze a jaké k nim patří UML diagramy) - pak napasovat na UP inkrementální a iterativní přístup. Agilní vývoj nejdřív obecně a pak SCRUM podrobněji. V-proces model testování, jak klasifikovat chyby (prý podle závažnosti). Příklady agilních metodik (XP, Scrum), jaký standard zvolit pro řízení projektu se SCRUM. Rozdíl mezi PMBOK a SCRUM. **Metriky založené na složitosti (na toto se ptá často a lidi to moc neví - check jestli to máme zde)**, cyklomatická složitost - na co je užitečná. Vysvětlit V-model a v které části se aplikuje cyklomatická složitost. Validace vs. verifikace. Vysvětlit typy testů, integrační postupy; Klasifikace metrik (procesní, produktové, zdrojové) + uvést příklady; Cyklomatická složitost - princip výpočtu, k čemu je dobrá a co ukazuje. LOC metriky - klady, zápory, specifika; Project cost calculation - COCOMO/COCOMO2 - co počítá, hlavní myšlenka, režimy (3 úrovně detailu, 3 vývojové módy, resp. 3 modely u COCOMO2); funkční body - k čemu, jak se počítají, jaké jsou jejich typy. Critical path method a PERT. > > **Šimková:** Gantt, milníky, nakreslit, CPM, PERT, PPP > > **Batko:** Procesy, životní cyklus projektu, plánování (PERT, CPM), závěrečná fáze (dokumentace, reklamace, hlášení chyb); Životní cykly IT projektů - vodopád, spirála, SCRUM a extreme programming. Trojúhelník time, scope, price. Jednotlivé role a jejich zodpovědnosti v SCRUMu. > [color=blue]

Projektové řízení

Každý standard ho definuje trochu jinak, ale spojuje je popis, že to je: - Set metod, technik, nástrojů, kompetencí, které se aplikují na projekty, aby splnili projektové požadavky, naplnili svůj cíl a pomáhají udržet rovnováhu mezi Cost(/Price)-Time-Scope. - Je dost citlivé na změny a rizika

CO JE PROJEKT

Každý standard má svou definici projektu, jsou odlišné, ale spojuje je to, že každý projekt směřuje k dodávce produktu. Projekt je tedy: - časově ohraničené úsilí (*temporary*), - které má unikátní vlastnost (*unique*), - dochází během něho ke změnám (*change driving*), - nachází se v něm určitá míra nejistoty (*uncertain*), - a ultimátně směřuje k dosažení cíle

1. Temporary - každý projekt má start, konec a svůj lifecycle (typicky pre-project, initiation, execution, closing): - **Pre-project**: studie proveditelnosti, příprava žádosti - **Initiation**: detailní harmonogram, rozpočty, sestavení týmu - **Execution**: vytvoření/realizace - není tak zajímavé pro project managera - **Closing**: předání, akceptace, fakturace, archivace a retrospektiva(lessons learned) ve firmě

2. Change driving - řízení změnami. Projekt má mít přínos (value - kvantifikovatelný benefit) pro stakeholdery (kdokoliv, kdo zasahuje nebo je zasáhnutý projektem = zaměstnanci/zákazníci/...). Přínos je popsán v business case.

3. Unikátnost - projekt není běžný byznys, není to rutinní splnění objednávky / dodávka balíčku. Kombinace jednotlivých faktorů ho dělá unikátním, ale každý projekt má repetitive elementy = procesy.

4. Uncertainty - Je to vlastně riziko, pravděpodobnost, že se stane událost, která způsobí škodu. Při projektech je hodně rizik, protože jde o novou činnost. Je potřeba počítat s riziky a připravit nějakou risk strategy, assessment apod.

Hranice projektu definuje **projektový trojimperativ** - rozsah, cena, čas (*scope, price/cost, time*). Změna jednoho parametru vede k změně *minimálně* jednoho dalšího. Např. chceme rozšířit scope, musíme zvýšit cenu i čas projektu. -



Co je projekt	Co není projekt
Vytváření nového IS	Instalace softwaru
Stavění domu	Opravení prasklé trubky
Přistání na Měsíci	Přistání na letišti

CO JE PROCES

Sled samostatných činností, které na sebe navazují a na základě vstupních požadavků vytvářejí výstup. Procesy jsou opakovatelné, lehce monitorovatelné a měřitelné, odladěné a vyzkoušené. Projekt je vlastně unikátní set procesů, které mohou být opakované a automatizované. Procesy mohou být používány v různých projektech.

Project vs Process

procedures which convert inputs into outputs [6].

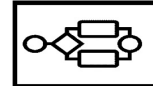
Project

- **Unique**
- Is only performed once
- Scheduled
- Managing a project means **keeping things on track**
- Best visualized with **Gantt chart**



Process

- **Continuously repeated**
- Has many instances
- Event driven
- Managing a process means **making things better and faster**
- Best visualized with **Flow chart**

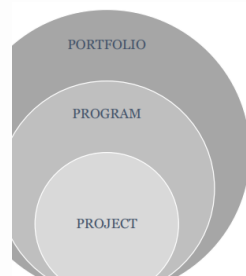


PPP - PORTFOLIO, PROGRAM, PROJECT

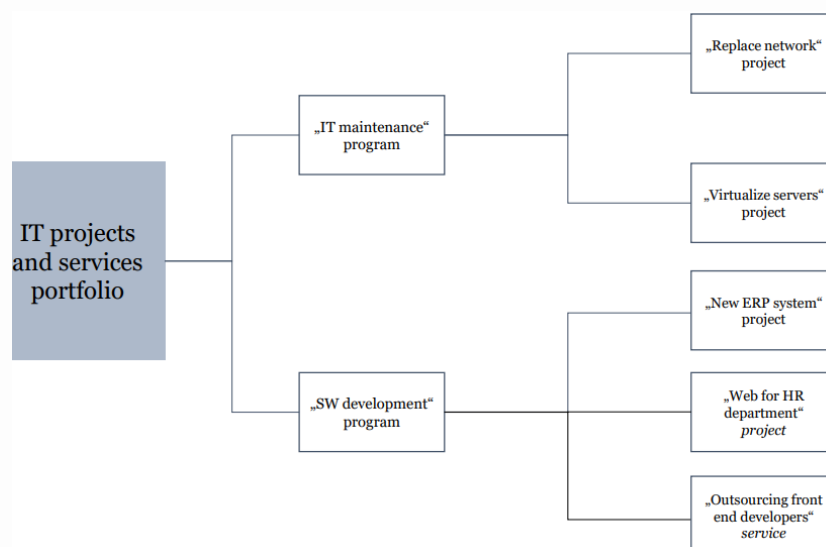
PROJECT - Dočasné úsilí vynaložené na vytvoření jedinečného produktu, služby nebo výsledku. Řídí je projektový manažer, který dohlíží na termíny, rozpočet a výsledek projektu. Co chceme vytvořit, za kolik peněz a kdy.

PROGRAM - Set dočasných projektů, které mají společné vlastnosti a směřují ke common objective - společnému cíli. **Přináší hodnotu stakeholderům** (*To deliver value to stakeholders*).

PORTFOLIO - Probíhající set projektů a programů, které spolu směřují ke strategic objective - strategickému cíli. **Přidává hodnotu businessu.**



Portfolio	Ongoing set of projects and programs managed as a group to achieve strategic objectives .
Program	Temporary, dependent set of projects managed as a group to achieve common objectives .
Project	Temporary endeavour undertaken to create a unique product, service or result.



Mezinárodní standardy projektového řízení

Využívají se, abychom dosáhli lepších výsledků, pracovali efektivně, zvýšili transparentci a "nevynalézali znovu kolo".

- **PRINCE2** = *Project in Controlled Environments*.
- **PMI PMBOK** = *Project Management Institute (PMI) Project Management Body of Knowledge*.
- **IPMA ICB** = *International Project Management Association (IPMA) Individual Competence Baseline*.

PRINCE2

Procesně orientovaná metodika - stávající verze je z roku 2017 (druhá verze). Má preskriptivní charakter, je to prostě vyplňování formulářů, papírů, nic pro kreativní lidi, protože to projektáka zdegraduje na správce excelovských tabulek. Jinými slovy je to *step-by-step formula* pro úspěšný projekt. Je procesně orientovaný a prakticky je to kuchařka, která vám ke každému kroku dá template dokumentu, který máte vyplnit. Projektákoví říká "jsi v téhle fázi a máš udělat tohle".

Staví na **7 základních principech**, - Learn from experience - "Capture and store lessons learned" - Focus on products - "The goal is not to work, the goal is to create the product" - ...

dotýká se z pohledu projektové řízení **7 témat**, - Business case - Why - Organization - Who - Quality - What - Plans - How, how much, when - Risk - What if - Change - What's the impact - Progress - Where are we now, where are we going, ...

a má **7 hlavních procesů** (které se rozpadají na další dílčí procesy.) - Starting up a Project (SU) - Directing a Project - Initiating a Project (IP) - Controlling a Stage - Managing Product Delivery - Managing a Stage Boundary (SB) - Closing a Project (SP) -

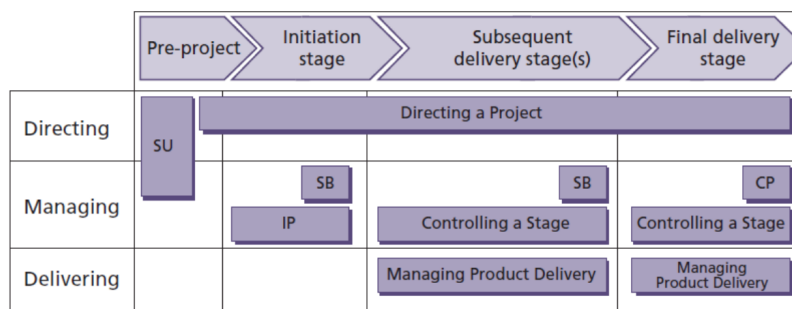


Image Source: Managing successful projects with Prince2, 2017 [2]

Who is responsible for the process:

Directing level - Project board (Executive, Project sponsor)

Managing level - Project manager

Delivering level - Team manager

Tuto metodu je nejlepší dodržovat od začátku do konce, vhodná je pro začínající i zkušené manažery. PRINCE2 nejčastěji využívají firmy, které potřebují důkladné reporty, detailní dokumentaci a management typu "kontrola a rozkazy" → například státní zakázky na ministerstvu, jaderné elektrárny. Ráček srovnával děláni e-shopu a softwaru pro jadernou elektrárnu. Pro e-shop je vhodné zvolit agilní metodiku typu SCRUM - naprogramuje výběr zboží, platební bránu, ale na komunikaci se zákazníky a monitoring skladu už nezbydou peníze. Kdežto u jaderné elektrárny je potřeba dodat úplně celý projekt, nemůže se nám stát, že naprogramujeme funkci zapnutí reaktoru, ale na naprogramování vypnutí reaktoru už nezbydou peníze. Metoda PRINCE2 se nezabývá managementem požadavků a rozpočtu (vhodné doplnit tuto mezeru pomocí přístupů jiných standardů).

PMBOK

Procesně orientovaná metodika (podobně jako PRINCE2), ale dává větší volnost. Je to správná metodika, pokud se rozhodneme, že začneme PRINCE2 a potom se zamyslíme, že ten projekt je menší a PRINCE2 je overkill (PA179-20190227 1:10:50).

Struktura PMBOK: -

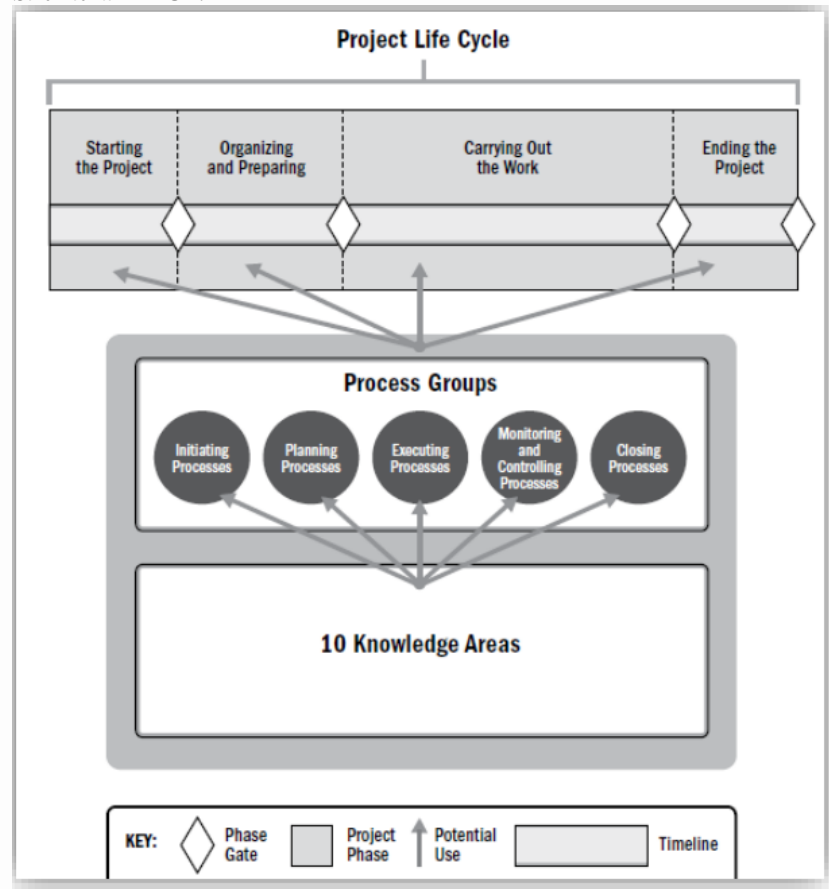


Image Source: A Guide to the Project Management Body of Knowledge, 2017 [3]

- **10 knowledge areas** (znalostní oblasti), každá oblast má své definované procesy:
 - Integration, Scope, Schedule, Cost, Quality, Resource, Communications, Risk, Procurement, Stakeholder

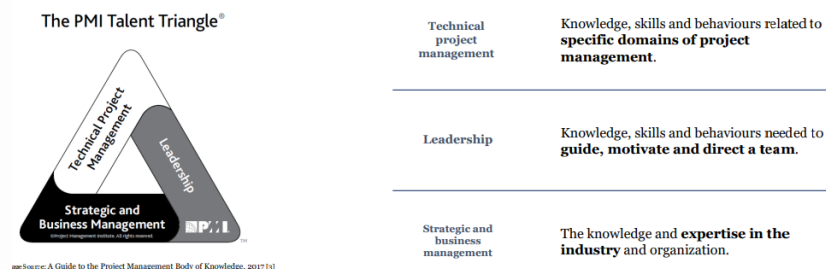
10 Knowledge areas and their main processes

Integration	Scope	Schedule	Cost	Quality
<ul style="list-style-type: none"> • Develop Project charter – basic information, base for applying resources • Develop Project management plan • Perform integrated change control – review, approve, deliver and communicate change requests ◦ • Close stage or project 	<ul style="list-style-type: none"> • Collect requirements • Define, validate and control scope • Create WBS 	<ul style="list-style-type: none"> • Define and sequence activities • Estimate activities duration • Develop and control schedule 	<ul style="list-style-type: none"> • Estimate costs and budget - of individual activities or work packages based on WBS • Control costs and budget 	<ul style="list-style-type: none"> • Plan, manage and control quality
Resource	Communications	Risk	Procurement	Stakeholder
<ul style="list-style-type: none"> • Estimate physical and team resources to all activities • Acquire and control resources • Develop and manage team 	<ul style="list-style-type: none"> • Plan, manage and monitor communications • Collect, create, distribute and monitor project information 	<ul style="list-style-type: none"> • Identify risks • Perform qualitative and quantitative risk analysis • Monitor risks • Plan and implement risk responses 	<ul style="list-style-type: none"> • Purchase or acquire products or services from outside the project team • Contracts, purchase orders, SLAs etc. • Select sellers and prepare contracts • Monitor contract performance 	<ul style="list-style-type: none"> • Identify stakeholders • Plan, manage and monitor stakeholder engagement – based on their needs

- **5 process groups:**

- Initiating processes
- Planning processes
- Executing processes
- Monitoring and Controlling processes
- Closing processes
- **49 procesů**, jsou rozdělené do 5 process groups a 10 knowledge areas:

K PMBOK existuje PMBOK process flow, což je diagram popisující každou process group a její procesy. Relativní novinka je PMI Talent Triangle, který reaguje na vlastnosti project managera - měl by mít hard skills, soft skills, strategic i business skills.



Nejllepší je používat tuto metodu jako guide, poradí jaké techniky a nástroje použít, může ho použít začátečník i pokročilý project manager. Certifikát z PMI je dost vyžadovaný v USA.

IPMA ICB (Individual Competence Baseline)

IPMA ICB je **Competence-based** approach. Její gró je takové, že popisuje jednotlivé kompetence (= „*Individual competence is the application of knowledge, skills and abilities in order to achieve the desired result.*“), které by měl projektový manažer splňovat. Je spíše pro pokročilejší manažery, kteří už mají nějaké zkušenosti.

Knowledge (pochopení Gantt chart) < **Skills** (vytvoření Gantt chart) < **Abilities** (schopnost úspěšně manažovat project schedule)

ICB má 3 oblasti kompetencí a pro každou jsou popsány požadované dovednosti a i způsoby jak je změřit: - 5 Perspective competencies - e.g., strategy, culture and values, compliance standard and regulation - 10 “People” competencies - e.g., leadership, teamwork - 13 Practice competencies - e.g., finance, stakeholders, project design

IPMA může být využita kdykoliv, je spíše používaná jako guide a obsahuje detailní sekci a soft skills.

"Tomato soup" PMBOK, PRINCE2, IPMA analogie: - PMBOK je jako kuchařská kniha šéfkuchaře - říká vám, jak vařit vodu, krájet cibuli a loupat rajčata - PRINCE2 jako recept na rajčatovou polévku - IPMA jako návod na získání šéfkuchařských kompetencí.

Procesní skupiny v projektu

information_source: [Zdroj](#)

V PMBOK jde o logické seskupení do inicializace, plánování, provedení, monitoring a kontrola a Closing (ukončení). Není to lifecycle, že bychom byli v "initiating phase", pak plánovali, pak prováděli a pak až kontrolovali.

Initiating phase Pomáhá nastavit vizi, čeho chceme v projektu dosáhnout. Všechny součásti, procesy a zdroje jsou identifikovány, abychom úspěšně dokončili projekt. Zahrnuje: - Identifikace účelu projektu: Tím, že víme k čemu projekt bude nám pomůže zjistit, co bude potřebovat - Identifikace stakeholders: Pro koho je projekt určen (kdo z něj bude benefitovat). - Navrhnout scope projektu: Spolu s projektovým týmem a stakeholders definovat rozsah projektu

Planning phase Vytvoření všech detailních dokumentů o projektu (milníky, rozpočet, požadavky, rizika, dokumentace, etc.). Jedná se o tzv. *projektový plán*. Zahrnuje: -

Vytvoření budgetu: Kolik bude projekt stát? - Risk assessment: Identifikovat potenciální problémy, než skutečně nastanou. - Project schedule a roadmap: Vzít vše, co víme z project scope a risk assessmentu, abychom vytvořili časový plán a roadmapu.

Execution phase Uskutečnění projektového plánu z předešlé fáze. Zahrnuje pravidelné meetingy s členy týmu.

Monitoring & controlling phase Pomáhá vyhodnocovat aktuální stav projektu a hledat řešení v případě odchylek od plánu (cena, rozsah, čas). Je aktivní po celou dobu projektu.

Closing phase Uzavření projektu, předání výstupů, podpis smluv, platby, shrnutí projektu atd. Good practise je i *lessons learned*, tzn. identifikace chyb pro poučení se do dalších projektů.

Životní cyklus IT projektů

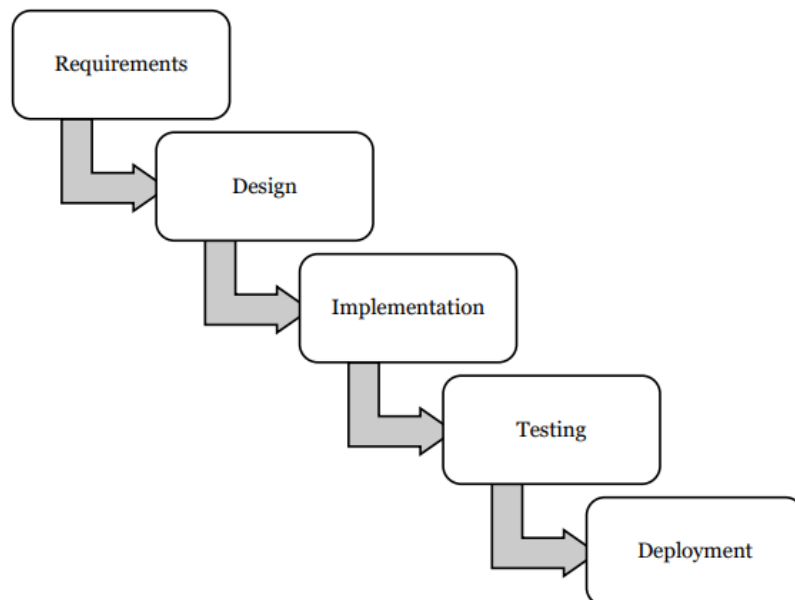
Rozdíl mezi standardním projektem a IT projektem je ten, že IT projekt používá delivery informační technologie. Jejich vývoj je rychlejší, IT projekty se často navzájem ovlivňují, rizika se nedají tak dobře předpokládat (potřebný risk management!) a tak se IT projekty typicky realizují **iterativně-inkrementálním** způsobem. Výsledek projektu je součástí služeb a produktů, které firma nabízí a vyžaduje další management.

ITIL - IT Infrastructure Library - popisuje best-practice pro IT service management, má 5 úrovní: - Service strategy - Service design - Service transition - Service operation - Continual service improvement

Životní cyklus IT projektů

Waterfall (vodopád)

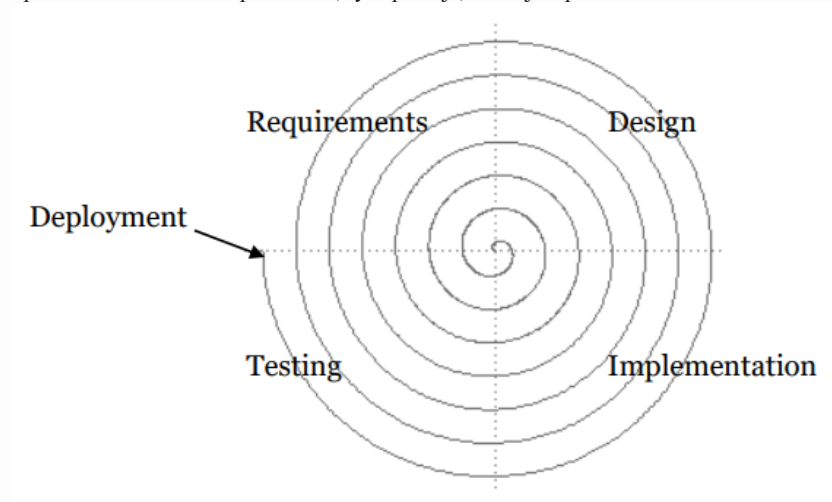
Je starý a nepoužitelný na SW development. Na začátku jsou požadavky, potom je zanalyzuju, implementuji, testuji a nasadím. Každá fáze se dělá pro celý systém (sbírám požadavky na všechny funkcionality, navrhují všechny funkcionality, etc.). Je to obrovské, neflexibilní, náročné na zapracování změn.



Spirála

Inkrementální (vždy přidáváme po trošce) a iterativní (designovaný v opakujících se cyklech). Vhodnější na SW development, většina momentálně používaných metod je nějaká variace spirálového modelu.

Začínám uvnitř spirály, nasbírám požadavky, navrhnu, naimplementuju, otestuju, upřesním/dám další vlnu požadavků, ty zapracuju, otestuju a pořád dokola.



Přístupy k SW developmentu

Prediktivní

Rigidní(nepružný, nepoddajný), zaměřuje se na procesy, má fixované požadavky, plánuje dopředu → například **Unified Process**.

Dost využívá UML, je to prediktivní, iterativně-inkrementální technika. Celý projekt se dělí z pohledu vývoje na iterace: - **Inception** - získáme požadavky - **Elaboration** - analýza, design, vytvoření UML diagramů - **Construction** - vytvoření - **Transition** - uzavření, testování odevzdání

Každá z těchto fází by neměla trvat víc než 3 měsíce, rozdíl mezi dvěma na sebe následujícími iteracemi se nazývá inkrement. Každá fáze musí obsahovat 6 workflows: - **Business Modeling** - Activity diagram - **Requirements** (Požadavky) - Use Case diagram - **Analysis & Design** (Analýza a design) - Class, Sequence, Collaboration diagram - **Implementation** (Implementace) - Class, Object, Component diagram - **Test** (Testování) - UC, Class, Activity diagram - **Deployment** - Deployment diagram

Používá se, když většina požadavků musí být specifikovaná dopředu, potřebujeme kompletní kontrolu nad procesy, týmy a detailní dokumentaci (ve formě UML)

Unified Process lifecycle:

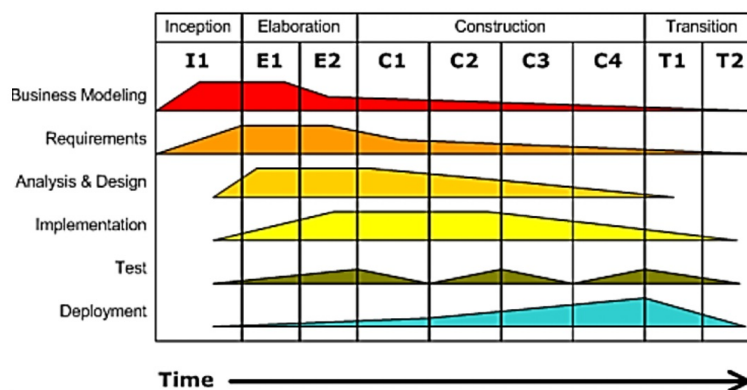


Image Source: Arlow, 2005 [11]

Agilní

Flexibilní, zaměřuje se na lidi, požadavky jsou pravidelně aktualizované, neplánuje moc dopředu → například **SCRUM**

Agilní, iterativně-inkrementální metoda. Teorie SCRUMU je popsána v "The SCRUM Guide", což je krátký dokument se striktními pravidly, které tvoří SCRUM tým, čím je. Očekává se, že další metodiky, techniky a nástroje jsou přidány, abychom dosáhli cíl projektu. SCRUM je charakterizovaný: - 5 Events - 3 Artifacts - 3 Roles

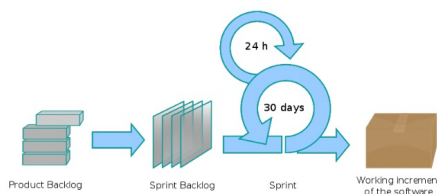


Image Source: Wikimedia Commons [3]

5 Events

Sprint
Sprint Planning
Daily SCRUM
Sprint Review
Retrospective

3 Artifacts

Product Backlog
Sprint Backlog
Product Increment

3 Roles

Product Owner
SCRUM Master
Team of Developers

ROLES

- *Product Owner* - Ten, kdo říká, co chce za své peníze. Vybírá si požadavky z *Product Backlogu* a co si vybere do následující iterace (*Sprint*), se nazývá *Sprint Backlog*. Na konci sprintu vzniká *Product Increment*. Vytvoření *Sprint Backlogu* se nazývá *Sprint Planning*.
- *SCRUM Master* - Je to vlastně projektový manažer. Zaměřuje se na procesy, jeho zodpovědností je dávat pozor na to, aby SCRUM zůstal SCRUMem a menežuje celý tento proces.
- *Team of Developers* - 3 až 9 lidí, udržují *Sprint Backlog*, jejich zodpovědností je vytvářet produkt.

ARTIFACTS

- *PRODUCT BACKLOG* - Reprezentuje celkovou specifikaci a odhaduje pracnost (technika na odhadování množství práce - Planning Poker). Jednotlivá funkcionality je vyjádřena ve formě *User Stories* a ty mají *Story Points*. Product Backlog vytváří celý SCRUM. Běžně je ve formě nějaké tabule, kde jsou sticky notes. Product Backlog se musí "nacenit", tj. udělit story points.
- *SPRINT BACKLOG* - Část User Stories z product backlogu, která definuje práci, kterou se plánuje vykonat v rámci sprintu - vytváří ho tým developerů. Vybrané User Stories jsou změněné do úloh a mají přiřazený časový odhad (story points). User Stories mohou být přidávány/odebrány, pokud se tím nenaruší Sprint Goal.
- *PRODUCT INCREMENT* - Reprezentuje všechny položky z Product Backlogu, které se vykonaly během Sprintu + minulý inkrement. Všechny položky musí být ve stavu "Done", ne v "in progress" ani "to-do".

EVENTS - *SPRINT PLANNING* - 2h meeting na začátku každého sprintu, kde se potká celý SCRUM tým a nastaví si Sprint Goal, vybírají si položky z Product Backlogu a přiřazují k nim úlohy. - *SPRINT* - Jedna iterace, která má vyústit ve funkční product inkrement, který může být releasnutý. Neměl by být delší, než 1 měsíc. Podílí se na něm celý SCRUM tým. Typický "analyse-design-build-test" přístup - *DAILY SCRUM* - 15min meeting týmu developerů každý den. Každý z členů odpoví na otázky: "Co jsem dělal včera? Co budu dělat dnes? Mám nějaký problém?" - *SPRINT REVIEW* - 2h meeting po měsíčním sprintu. Přítomný je celý tým a klíčoví stakeholders. Překontroluje se inkrement, zhodnotí se, projde se product backlog, spočítá se progres a přepočítají se daty. - *SPRINT RETROSPECTIVE* - 3h meeting za 1 měsíční sprint. Je tam celý SCRUM tým, diskutuje se aktuální sprint, jeho členové, vztahy, procesy a nástroje. Cílem je vymyslet aspoň jedno vylepšení do následujícího sprintu.

SCRUM se může ukončit např.: - Product backlog je prázdný - Nejsou peníze, čas, ... -

Product owner se rozhodne zastavit projekt

SCRUM se používá, když přesné požadavky nevíme dopředu. Tým dobře spolupracuje a komunikuje a zákazník chce nějakou formu produktu ASAP.

Plánování projektu

Při plánování projektu se vybere přístup k managementu. Musíme vzít do úvahy různé věci, například velikost týmu, délka projektu, komunikace a kooperace pro time, delivery (chceme produkt co nejdřív, klidně i ve verzi, která se bude ještě upravovat, tzn. prototyp?), typ požadavků (budou časem přibývat, nebo je známe všechny na začátku?)

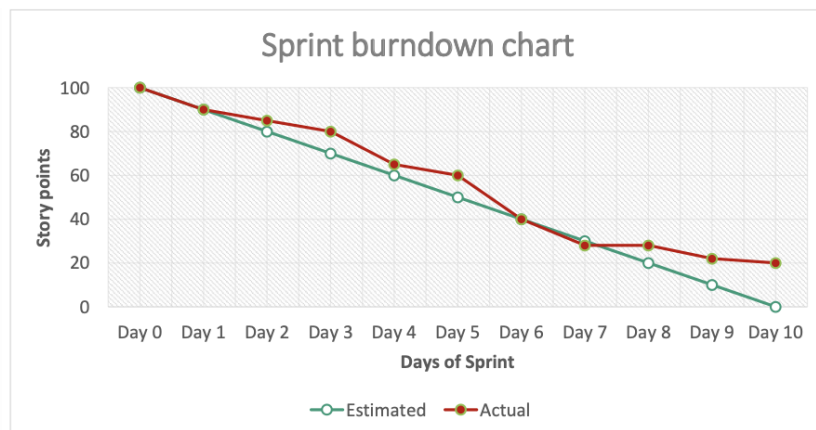
Agilní přístup - IS pro disabled students

> PA179 04_AgileExample01 až 07_AgileExample04

Skládá se z 3 částí: - Planning - Sprinting - Closing

PLANNING

Můžeme konzultovat IPMA ICB a PMBOK, které nám pomohou s formální stránkou projektu: - **Vytvořením Projektového Charteru** - Definice: Dokument, který formálně autorizuje existenci projektu a poskytuje projektovému manažerovi autoritu použít organizační zdroje na aktivity projektu. - Poskytuje ho IPMA. Má 5 otázek: - BUSINESS CASE = WHY - vysvětlíme oficiální důvody(proč se to dělá), benefity projektu z pohledu firmy (skoro jako marketing). Načtneme největší risky, výdaje a budget. - OUTCOME = WHAT - jednoduchý popis produktu, požadavky, hlavní cíle - STAKEHOLDER = WHO - identifikace klíčových externích+interních stakeholderů - APPROACH = HOW - popis jak bude řízený projekt - lifecycle, použité standardy, vybrat vhodné nástroje, metody, koncepty. - SCHEDULE = WHEN - klíčové milníky - Management klíčových strategií (komunikace, risks, cena, kvalita, kompetence, changes) - *Communication management strategy* - Nastaví si, jak se bude komunikovat - face-to-face, dokumentace, show-don't tell, rychlé meetingy. Je potřeba krátce sepsat, jaké budou meetingy, reporty, kdo se jich bude účastnit a jaké budou užívané mechanismy. - *Risk management strategy* - Vytvoření risk strategy - která se typicky skládá z (v IPMA, PMBOK): - Identifikování základních rizik - Ohodnocení rizik - hodnocení = pravděpodobnost, že se projeví * dopad (nutno definovat následky) - Odpovědi na jednotlivé risky - Transfer/Avoid/Reduce/Accept - Specifikace monitoringu risků - kdo je zodpovědný? Kde budou risky definovány, reviewed? - Vytvoření registru risků - Mezi nejčastější rizika patří nedostatek lidí, nerealistické odhady cen a času, nepochopení požadavků. - Co snižuje rizika? - Transparentnost co se děje, feedback, používání user stories (popis toho, co má být uděláno), začlenění zákazníka do plánování, krátké iterace. - *Change management strategy* - Jak se bude postupovat, když zákazník přijde se změnou během vývoje. Při novém požadavku se rozhoduje, jestli se zastaví nebo nezastaví aktuální sprint. - *Quality management strategy* - Otázky ohledem kvality řízení procesu, kde si stanovíme quality objectives (e.g., deliver on-time s maximální 2% chybovostí), míry, které zlepšují procesní kvalitu (nástroje). - V SCRUM např. User stories s akceptačními kritérii, sprint retrospectivy, burndown chart (měří schopnost týmu přinášet inkrementy včas. Osa x reprezentuje čas od začátku sprintu/projektu a osa y zbývající story points = imaginární jednotka pracnosti). Team velocity = kolik story points dokáže tým dokončit za den (sprint).



- Management *Product backlog* Do product backlog se dávají user stories (1), které ohodnotíme z pohledu pracnosti (2) a dáme jim priority (3)
 - User stories popisují funkcionalitu z pohledu uživatele, součástí jsou i akceptační kritéria (co znamená, že požadavek je splněný), a možné související rizika.
 - Odhadování pomocí *planning poker* (hlasuje se s kartičkami, dokud se nepřijde k společnému číslu. Pokud se lidé neshodnou, musí svoje volby okomentovat) na pracnosti user stories.
 - Prioritizování pomocí MoSCoW metody (must have, should have, could have, won't have)

SPRINTING

- Sprint planning - z product backlog se něco vybere a to se stane Sprint goal
- Daily SCRUM - Podporuje transparentnost, progres.
- Sprint review - Zaměřený na nový inkrement, co se má udělat, co se udělalo, kontrola product backlogu. Měří progres.
- Sprint retrospective - Zaměřený na tým, komunikaci, nástroje a procesy. Měří výkonnost (performance).
- Updating key strategies - Pokud se zjistí, že se něco dělá špatně, tak se updatnou strategie.

Do samotného sprintu patří i:

- Řešení týmových konfliktů
 - Fáze konfliktů:
 - Latent (možné, zatím nepozorované)
 - Emergent (viditelné, zatím rozumné)
 - Escalated (open conflict)
 - Můžeme použít např. De-escalation technique, kde zmírníme nejdřív emoce, jdeme si v klidu pokecat, aktivně nasloucháme a hlavně komunikujeme s respektem a nesoudíme
- Změna požadavků - jediná věc, kterou je potřeba rozhodnout je to, jestli se sprint zastaví nebo ne

CLOSING THE PROJECT

- Finišování produktu - na konci se běžně dohodneme na posledním sprint review
- Přechod k operations, dokumentace - uživatelská dokumentace je vyráběna průběžně, technická dokumentace záleží na dohodnuté formě supportu (více o tech. dokumentaci hovoří ITIL)
- Timeline retrospective - Co jsme se naučili? Co se mělo stát a co se doopravdy stalo?

PREDIKTIVNY PRISTUP - mestske kiosky

> PA179 08_PredictiveExample01 - 11_PredictiveExample04

V príklade kombinujeme PRINCE2 (pomaha s terminologiou, rolami, základnymi princípmi,

reporting, projektovou dokumentaciou, projektový plan => cize s riadenim projektu) a unified process (pomaha s procesom vyvoja produktu, zberom požiadavkov, produktovou dokumentaciou => cize s vyvojom produktu). Manazment bude velmi administrativny s detailnou dokumentaciou, praca je kontrolovana a udavana, scope definovany dopredu.

STARTING PHASE

- Project brief (PRINCE2) - zakladny dokument, prakticky rovnake ako project charter + definovat tim, role + definovat aky manazmentovy pristup zvolime
- High-level requirements (UP) + definovat viziu, core funkcionalitu
- Feasibility analysis (UP) + odhad ceny, casoveho planu, riskov
- Outline architecture (UP) - navrhne aspon 1 mozne riesenie
- Next stage plan (PRINCE2) - vytvorit detailny day-to-day plan => WBS, identifikacia aktivít, odhadutie ich trvania, milestones. Vytvorit Gantt/Network diagram

INITIATION = ELABORATION PHASE

- *Project Initiation Documentation (PID)* (spisane zakladne principy projektařiny - spravi sa detailny business case, rozpracovat detailnejšie strukturu timu, change, quality, risks, communication management approach ... je to vlastne detailny project brief) + jeho sucastou je *Project plan* (PRINCE2)
 - Detailny business case definite dovody na projekt, ocakavane benefity, analyzu výdajov a príjmov, najvacšie risky, ...
 - Struktura timu - Project board (Executive = zadavatel + Zastupca dodavateľa + Zastupca koncového užívateľa) a pod nimi su projektovi manazeri, timovi manazeri. Project board schvaluje všetky plány, výnimky a odchylky od planu
 - Quality management approach - vytvára sa Quality register (zaznamy o všetkých aktivitách, ktoré sa maju robiť), quality manager kontroluje napr. či sa dodržiavajú ISO standardy, ako často sa testuje, čo sa testuje
 - Change control approach - vytvára Issue register (definícia čo sa spravi, ak pride požiadaviek na zmenu)
 - Risk management approach - vytvára sa Risk register (identifikujeme rizika, planujeme opatrenia)
 - Communication management approach - definovat napr. Kto komu čo posielá, ake reporty, ako často
 - *Project plan* - vytvorenie WBS (popísané nižšie), vypocitat casove a cenove odhady (PERT, popísané nižšie), identifikovanie závislosti a odhadnutie aktivít (Precedence diagramming method, popisane pri agilnom projekte), rozvrh (Gantt chart vyššie + Network chart nižšie)
- *Detailed requirements* (UP)
 - Zhromaždit detaily vacšiny požiadavkov
 - Vytvorenie detailných use-casov na vacšinu funkcionality (zacínajú tu pracovať analytici)
 - Vytvorenie špecifikácie produktu - dvojita špecifikácia, jednak use-cases pre vyvojárov a jednak scenare pre užívateľov (tu kliknes a zobrazí sa obrazovka). Produktová špecifikácia je základom pre kontraktovanie (zakazník platí až na konci za celý produkt)
- *Stage report* (PRINCE2) - report za prave prebehnutú etapu, popisuje progres, celkovú situáciu projektu, ohodnocuje timový performance, produkt a jeho kvalitu, obsahuje súhrn aktuálnych riskov, problémov
- Next stage plan (PRINCE2) - rovnake ako pri starting phase

DELIVERY PHASE

Zvyčajne je komponovaná z viacerých fáz, každá je ako malý projekt a nemala by trvať viac ako 3 mesiace. V tejto fáze je rola projektového manažera udržať projekt *cas, budget, scope a kvalitu tak, ako bolo dohodnuté v Project Initiation Documentation*. Zaciná už analyzou, dizajnom, pokračuje programovaním a konci testovaním, integrovaním. PRINCE2 nerozlišuje praktickú implementáciu ako delivery narozdiel od UP, ktorý považuje za delivery práve tu praktickú implementáciu.

- *Robust Architecture* (UP)
 - V UP je to este sucasť elaboration ale PRINCE2 to už považuje za delivery keďže ide o návrh architektúry - moduly, technologie, identifikácia rizík (napr.

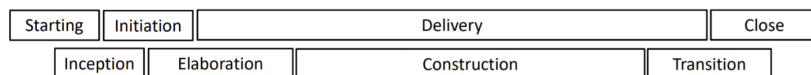
- kompatibilita, výkon HW)
- *Controlling a stage* - projektový manažer vedie dokumentáciu v ktorej popisuje, ako vyvoj vyzerá. Obsahuje dokumenty ako:
 - Work package - popis činnosti, čo sa práve robí, kto to robí, kým sa akceptuje, ...
 - Highlights report - súhrn o fáze, ktorý sa predáva project board
 - Daily log
 - Issue/Risk register updates - zmeny požiadavky i rizika sa chovajú podobne, treba na to reagovať. Ak požiadavka alebo riziko presiahne toleranciu/projektu => eskalácia na project board a vyplní sa Exception report => na neho sa viaže Exception plan, ktorý pokrýva čas od teraz do konca fázy
- *Managing product delivery* - z pohľadu tímového manažera
 - Team plan - dokument, ktorý hovorí kto bude na čom pracovať
- *Managing a stage boundary* = kontrolovanie hraníc jednotlivých fáz vykonávanie projektovým manažerom
 - End stage report - ako stage report v initiation phase
 - Next stage plan - ako pri starting phase
 - Project plan update
 - Risk register/Issue register/Quality register update
 - Business case update - to sa obvykle nedeje

TRANSITION (UP) AND CLOSING (PRINCE2) PHASE

Transition phase: - *System acceptance (UP)* - Acceptance existuje v UP pretože PRINCE2 je metóda pre rôzne typy projektov, nie len IT. V poslednej fáze Construction v UP máme k dispozícii už Beta system - relatívne stabilný system, integrovaný, testovaný, pripravený na užívanie ale beta verzia sa používa 2-3 mesiace a potom sa prejde na produkciu - *Transition phase (UP)* - Cieľom je urobiť akceptovateľný system, aby sa dal používať - Dopisuje sa dokumentácia - užívateľská, vývojárska, operations, trainings - Začína training užívateľov, ktorí budú so systemom pracovať - Beta testing - prináša user feedback čiže vlastné požiadavky na zmeny, treba si vymyslieť spôsob spracovania a zbierania feedbacku. V tomto bode je dôležitý *Configuration management* pretože je bez neho máť aj 3 rôzne prostredia (production, testing, dev) - *Product acceptance testing* je formálne testovanie systému, finálna verifikácia požadovanej funkcionality, ktorá overí či boli splnené všetky požiadavky kontraktu. Vypisuje sa dokument *Acceptance protocol*

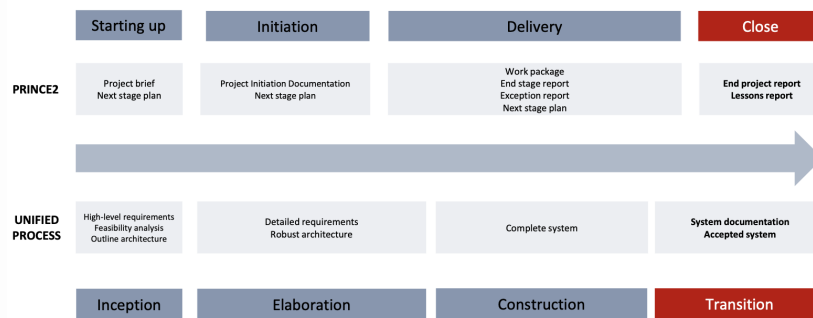
Closing phase: - *Product handover (PRINCE2)* - Čo bude ďalej so SW - definujeme SLA = Service level agreement, zmluva o servisnej podpore, kde sa definuje ako sú typy chýb, reakčné doby, pokuty, cena servisu, ... SLA je súčasťou Service Level Managementu, ktorý je súčasťou ITIL Service Designu - Presunutá zodpovednosť za produkt z projektu na operations a údržbu - *Zatvorenie všetkých registrov, logov, archivácia dokumentácie* - *End project report (PRINCE2)* - formálny report s "tvrdými" číslami - *Lessons learned report (PRINCE2)* - menej formálne, skor pred tým. Doporučenia do budúcnosti, pohľad zákazníka, nečakané udalosti, review užitočných techník

PRINCE 2



Unified Process

The Project's lifecycle - deliverables



Výpočet nákladů na projekt

Například pomocí PERTu, kde od pracnosti jednotlivých tasků vypočítáme odhadovaný čas potřebný na dokončení práce.

Existuje více způsobů, jakým se vytváří budget, jeden z nich je: - Určí se cena všech aktivit. Z PERTu víme, kolik budou stát jednotliví lidé. K tomu přidáme další výdaje - SW licence, cestovné, ceny nástrojů a materiálů, PR, HR, ... - Traverzováním WBS zdola nahoru určíme cenu každého work package a z toho vyvodíme cenu nazvanou *Project estimate* - K *Project estimate* se přidává *Contingency reserve*, která reprezentuje sumu všech projektových rizik. *Project estimate* a *Contingency reserve* spolu tvoří *Cost baseline* - číslo, které se používá na určení CPI / CV. - Na vrch *Cost baseline* se přidává *Management reserve* na zatím neurčené výdaje - Toto vše dohromady tvoří *Cost budget*

PERT (Program Evaluation and Review Technique)

> PA179 09_PredictiveExample02 > PA179 PA179-D2-20190410.mp4 - 1:05:30-1:19:35

Pravděpodobností technika pro odhadování potřebného času ke splnění tasku (naprogramování nějaké feature, atd.) Jednomu úkolu přidělíme tři typy potřebného času: - **Optimistický:** *o* (*minimum possible*) - **Pesimistický:** *p* (*maximum possible*) - **Nejpravděpodobnější:** *m* (*best estimate*)

Následně z těchto času vypočítám *očekávaný čas te*: - $te = (o + 4m + p)/6$

Tímto získám očekávaný čas v MD se schovaným "bufferem" (kdyby se něco pokazilo a task by trval déle času). Normálně je to poměr "1+4+1 / 6", ale můžeme si nadefinovat i své poměry. Jakmile získám očekávanou cenu tasku, přiřadím si k nim role a kolik na daný task spotřebuje MD. Dokážu si tam i spočítat "dobu trvání" - záleží, kolik času člověka alokuju na daný task, například task *Request Analysis* potřebuje 4.5MD, ale bude mi trvat 6 dní, jelikož přiřazený analytik bude během těchto 6 dní i pracovat na jiných věcech.

Work package	Time estimates in MD			Expected effort (te) in MD	Wage level MD	Effort per wage level (MD)	Expected duration (work days)
	Optimistic (o)	Most likely (m)	Pessimistic (p)				
1.2.1.1. Request Analysis	3	4	8	4.50	Analyst	4.50	6
1.2.1.2 Layout design	8	10	20	11.33	Designer	8.11	10
					Architect	3.22	
...

Síťový analýza, Kritická cesta

Síťová analýza

Síťová analýza se využívá hlavně na zobrazení toku peněz a času v projektu. Na síťovou analýzu se používá i Gantt chart, který může zobrazovat kritickou cestu.

GANTT CHART A PROJEKTOVE MILNIKY

Ganttův diagram je esenciální nástroj na projektové plánování. Reprezentuje plánované aktivity ve formě "kalendáře" (os Y reprezentuje aktivity, osa X datумы), který obsahuje jejich start, konec a závislosti mezi nimi. Může být vylepšen o milníky (důležité body delivery) a zdroje (kdo co dělá).

Součástí Gantt chart je často Precedence Diagramming Method, používaná na zobrazení možných závislostí mezi projekty:

- Finish-to-start - Aktivita B nemůže začít, dokud neskončí aktivita A.
- Finish-to-finish - Aktivita B nemůže skončit, dokud neskončí aktivita A.
- Start-to-start - Aktivita B nemůže začít, dokud nezačne aktivita A.
- Start-to-finish - Aktivita B nemůže skončit, dokud nezačne aktivita A.

Kritická cesta

Metoda, která pomáhá s kontrolou, jestli stíháme projekt dokončit včas. Vypadá jako síťový diagram, který definuje aktivity a doby jejich trvání. Do grafu se pomocí dopředného a zpětného průchodu dopisují:

Dopředný průchod - Určuje časy brzkého začátku (**Early Start - ES**) a brzkého konce (**Early Finish - EF**) pro každou úlohu. - Postupuje se zleva doprava. - Přidávají se časy na každé cestě. - Pravidlo: pokud několik úloh konverguje do jednoho uzlu, pak čas **ES** následující úlohy je roven největšímu z EF časů předchozích úloh. - Pomocí dopředného průchodu počítáme nejdřívější možné začátky a konce projektu. Začínáme v bodě 0, ak vidim ze trvanie = 10, zacnem v 0 a skoncim najskor v 10ty den. Aktivita F,B,H mozu zacat najskor v 11. F trva 15, cize najskor moze skoncit v 25. Na vypocitanie konca pripocitame DUR ku zaciatku, v pripade ak ide o vazbu Finish-to-start (prenasam koniec na start), ktora je najcastejsia. Start-to-start by preniesol start z A do startu B,F,H (1->1) , finish-to-finish by preniesol koniec aktivity A na koniec aktivit B,F,H a musim dopocitat start. Je to dobre pre alokaciu zdrojov.

Zpětný průchod

- Určuje čas nejpozdějšího konce (**Late Finish - LF**) a čas nejpozdějšího začátku (**Late Start - LS**).
- Začíná se v koncovém uzlu.
- Vypočítají se spodní páry čísel.
- Odečte se trvání od času brzkého konce v připojeném uzlu
- Pravidlo: pokud několik úloh konverguje do jednoho uzlu, pak čas LF následující úlohy je roven nejmenšímu (opačně jako u dopředného průchodu) z LS časů předchozích úloh.
- Pomocou zpětného průchodu počítáme najneskorsie mozne zaciatky a konce. Po vypocitaní dopředného průchodu postupujeme od poslednej aktivity a odratavame DUR (E 65-20=45, na 46 den najneskor musim zacat poslednu aktivitu). G,D,H musi najneskor skoncit na 45ty den aby posledna aktivita mohla zacat hned o den neskor. H musi zacat najneskor na 31. Den (45(skenci v tento den)-15(DUR)=30, cize nasledujuci den). Tu vidime, ze H moze zacat najskor na 11ty den a najneskor na 31.den, inak sa projekt omeska.

Výsledná Critical Path - Cesta poskladaná z uzlov, kde **EF** a **LF** sú rovnaké.

Objavi sa linia, kde najskorsie a najneskorsie mozne zaciatky a konce su rovnake. Tu sa nesmiem omeskat, pretoze potom sa cely projekt omeska a to je kriticka cesta.

Optimalizacia mimo kriticku cestu NEMA vplyv na cele trvanie projektu, cize skratenie projektu ma zmysel hladat v aktivitach v ramci kritickej cesty. Po ich skrateni ale moze nastat kriticka cesta inde!

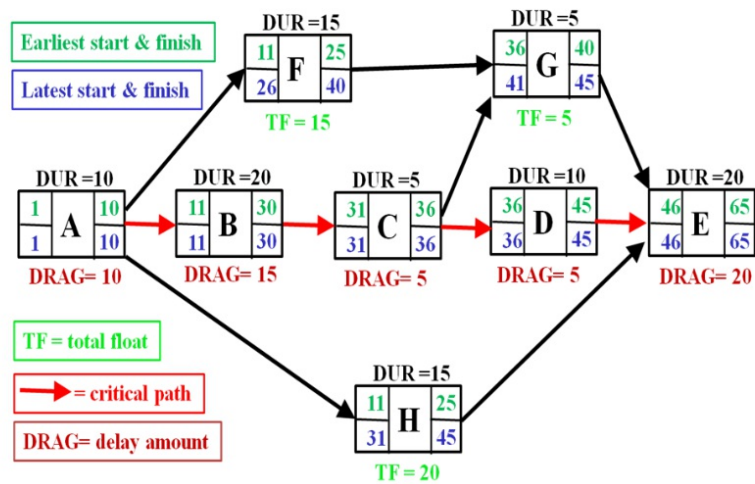


Image Source: Wikimedia Commons [6]

WBS

NIE JE súčasťou hlavných otázok. Je to hierarchická dekompozícia práce, ktorá sa má vykonať na celom projekte. Celý projekt dekomponujeme na časti= Work packages, čo sú najmenšie dekomponovateľné jednotky. Tie sa môžu skladať z ďalších činností ale sú už tak malé, že nemá zmysel ich ďalej rozoberať.

Pravidlá pre tvorbu WBS: - Dekomponujeme celý projekt, v súčasnosti by celý diagram mal vyjadrovať práve 100% - Každá činnosť by mala byť zahrnutá v celom diagrame práve 1 krát

1. City kiosk SW

1.1 Project management

1.1.1 PID

1.1.1.2 Project Plan

...

1.2 Transport support subsystem

1.2.1 Timetables

1.2.1.1 Requests analysis

1.2.1.2 Layout design

1.2.1.3 Prototype creation

1.2.1.4 System implementation

1.2.1.4 DB integration

1.2.1.5 System validation

1.2.4 Parking tickets module

...

1.3 City Hall directory

...

1.4 News and events module

...

ZAISTENIE KVALITY PROJEKTU, TESTY, PRESKUMANIE, MERANIA A STANDARDY

Všetky tieto veci boli nejakým spôsobom popísané pri agilnom a prediktívnom prístupe. Tu uvádzam informácie, ktoré neboli v prednáške ale v ISe v interaktívnej osnove.

Vhodne pozriť:

<https://docs.google.com/document/d/1VdDZRrxK3bJLfXBRhOBmjYBoZ4FMElal-IsUGigI9w/edit#heading=h.ljn1iwwgsfb>

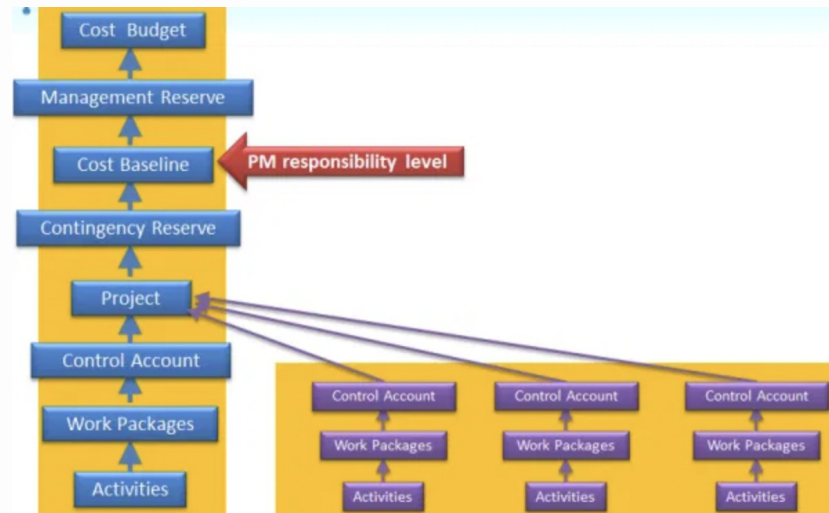
<https://www.projectsmart.co.uk/project-management-scope-triangle.php>

MERANIA

Projektovi manažeri sú vždy nejakou formou kontrolovaní - či už ide o zákazníka, upozorení alebo vrchný manažment. Títo ľudia sa môžu zúčastňovať stretnutí a byť tak v obraze no častejšie vyžadujú iba reporty o statuse. Zvyčajne sa používajú 4 metriky: - SPI = Schedule performance index - vyjadrené ako pomer množstvo zatiaľ urobenej práce / množstvo práce, ktorá bola plánovaná, že bude doteraz urobena. $SPI > 1$ znamená, že sme popredu, $SPI < 1$ znamená, že meškame - SV = Schedule variance - rovnake ako SPI ale hodnoty sa odčitajú. Výsledok < 0 znamená, že meškame - CPI = Cost performance index - koľko budget sa zatiaľ minulo v porovnaní s hodnotou, čo zatiaľ vzniklo. Interpretácia je analogická ku SPI - CV = Cost variance - rovnake ako CPI, ale hodnoty sa odčitajú

Pri standardoch sa dáva pozor na to, či sa dodržiavajú napr. ISO normy, platná legislatíva,

GDPR, ...



ZAISTENIE KVALITY

So zaistenim kvality suvisi hlavne hrozba riskov, ktore moznu projektu znacne poskodit. V risk managemente rozlisujeme kvalitativnu a kvantitativnu cast ohodnocovania riskov. - Kvalitativna cast - Najprv definujeme vsetky mozne udalosti, ktore sa mozu udiať. Tieto udalosti nemusia byt iba negativne (=threat), mozu mat ja pozitivny dopad (=opportunity). Kazdy risk ma priradene 2 dimenzie na ohodnotenie pravdepodobnosti a dopadu daneho risku na projekt. Obe hodnoty su v istej mierke, napr. 1-5, kde nizsia hodnota znaci mensiu pravdepodobnost alebo mensi dopad na projekt. Tieto 2 hodnoty sa potom vynasobia a ziskame konecnu hodnotu, ktora nam stanoví ako velmi by sme sa mali zaujimat o dany risk

5x5 RISK MATRIX						
PROBABILITY	Highly Probable	5 Moderate	10 Major	15 Major	20 Severe	25 Severe
	Probable	4 Moderate	8 Moderate	12 Major	16 Major	20 Severe
	Possible	3 Minor	6 Moderate	9 Moderate	12 Major	15 Major
	Unlikely	2 Minor	4 Moderate	6 Moderate	8 Moderate	10 Major
	Rare	1 Minor	2 Minor	3 Minor	4 Moderate	5 Moderate
		Very Low	Low	Medium	High	Very High
IMPACT						

- Kvantitatívna časť
 - Všetky pravdepodobnosti sa prerátajú do percent (ak sa incident I stane raz za 10 rokov a projekt je plánovaný na 2, pravdepodobnosť je 20%). Tiež sa transformujú “abstraktne dopady” na niečo konkrétne, najčastejšie na napr. zarobene/utratené peniaze navyše.

Na risky treba nejak zareagovať, a teda sa planuje *Risk response plan*, ktorý hovorí o tom, čo s daným riskom urobíme. Rozlišujeme 4 kategórie:

- Avoid / Exploit (pri pozitívnych)
- Mitigate / Enhance (pri pozitívnych)
- Transfer / Share (pri pozitívnych)
- Accept

Ako ďalšiu položku v manažmente riskov vytvárame *Contingency plan*, ktorý hovorí o akciách vykonaných ak risková udalosť nastane. Risk management je nepostrádateľný pre

uspesneho projektoveho manazera, vďaka tomuto manažmentu je tím pripravený na možné udalosti a vie na nich promptne reagovať.

TESTING

Unit testy => Komponentové testy => Integrované testy => Test celého systému

Acceptance testing